Vol. 37 No. 10 Transactions of Information Processing Society of Japan Oct. 1996

*Regular Paper*

# An Approximation Algorithm for MAX 3SAT

TAKAO ONO,† TOMIO HIRATA† and TAKAO ASANO††

In this paper we present a 0.769-approximation algorithm for MAX 3SAT. It is a restricted version of MAX SAT, in which every clause has at most three literals. The best previously know approximation algorithm for MAX SAT had 0.755-approximation ratio, which was given by Goemans and Williamson. Thus, we make a slight improvement by limiting MAX SAT to MAX 3SAT. Since approximating MAX 3SAT within 38/39 is NP-complete, our result means that the best approximation ratio is between 0.769 and 38/39.

## 1. Introduction

We consider the Maximum 3-Satisfiability Problem (MAX 3SAT), a restricted version of the Maximum Satisfiability Problem (MAX SAT). For MAX SAT, 0.75-approximation algorithms and 0.755-approximation algorithm were proposed. Yannakakis[10] first gave a 3/4-approximation algorithm for MAX SAT. Goemans and Williamson[5] gave another 3/4-approximation algorithm, and reported a slight improvement of the ratio to 0.755 later[4]. But no good approximation algorithms for MAX 3SAT are known. In this paper we present a randomized 0.769-approximation algorithm for MAX 3SAT. Thus we make a slight improvement by limiting MAX SAT to MAX 3SAT.

Since MAX 3SAT is MAX SNP-complete[8], for some constant $\alpha < 1$ no $\alpha$-approximation algorithms for MAX 3SAT exist unless P = NP. In fact, approximating MAX 3SAT within the approximation ratio 38/39☆☆ is known to be NP-complete[3]. Our result means that the best approximation ratio is between 0.769 and 38/39.

Our result is inspired by the recent work of Goemans and Williamson[4] who used an algorithm for semidefinite optimization problems to obtain an approximation algorithm for MAX 2SAT. Their algorithm is 0.878-approximation. But in the case of MAX 3SAT, we have to treat polynomials of degree 4 instead of those of degree 2. In our algorithm we carefully truncate some portion of the polynomials. Furthermore, we combine this algorithm with other known algorithms to obtain a good approximation algorithm.

In Section 2, some definitions are given. We present an approximation algorithm for MAX 3SAT based on the semidefinite programming in Section 3. In Section 4, we analyze the approximation ratio of the algorithm. Combining this algorithm with Johnson's algorithm[6] and LP-relaxation algorithm[5], we show that the 0.769-approximation ratio can be achieved in Section 5. Conclusion with some remarks is given in Section 6.

## 2. Definitions

Let $\{x_1, \ldots, x_n\}$ be a set of variables. A literal is either a variable $x_i$ or its negation $\bar{x}_i$. A clause is a disjunction of literals. An instance $I$ of MAX 3SAT is a collection of clauses $C_1, \ldots, C_L$ with the corresponding positive weights $w_1, \ldots, w_L$, where each clause has at most three literals. We assume that no variables appear more than once in a clause, that is, we do not allow a clause like $\{x_1, \bar{x}_1, x_2\}$. We denote by $|C_j|$ the number of literals in $C_j$. $I_j^+$ and $I_j^-$ denote the sets of indices of the variables that appear positively and negatively in $C_j$, respectively. The weight of an assignment is the sum of the weights of the clauses which are satisfied by the assignment.

For an algorithm $A$, $W_A(I)$ denotes the weight of the assignment produced by $A$ for an input $I$. The approximation ratio of $A$ is $\alpha$ if $W_A(I)$ is at least $\alpha$ times the weight $W_{OPT}(I)$ of the optimal assignment for any instance $I$. We call $A$ an $\alpha$-approximation algorithm. We use the notation $f(\boldsymbol{x})$ for multivariate func-

† School of Engineering, Nagoya University
†† Department of Information and System Engineering, Chuo University

☆ This is a revised version of Ref. 7), in which we erroneously claimed that our algorithm achieves 0.80-approximation.

☆☆ At present, the ratio is reduced to 26/27 [2].

tion $f(x_1, \cdots, x_n)$.

## 3. An Approximation Algorithm

In this section, we present an approximation algorithm for MAX 3SAT. This algorithm solves MAX 3SAT in the same way as Goemans and Williamson's algorithm for MAX 2SAT. The algorithm consists of the following steps.

(1) Translate an instance of MAX 3SAT into an integer programming problem.

(2) Reduce the problem to a semidefinite programming problem with relaxation and solve it.

(3) Construct a truth assignment of the original MAX 3SAT problem from a solution to the semidefinite programming problem.

We explain each of the above steps in the following subsections.

### 3.1 Translation of an Instance into an Integer Programming Problem

We arithmetize the clause $C_j$ as

$$C_j(\boldsymbol{x}) = 1 - \prod_{i \in I_j^+} (1 - x_i) \prod_{i \in I_j^-} x_i.$$

Thus, we can formulate MAX 3SAT by the following integer programming problem:

$$\begin{aligned} &\text{Maximize} && \sum_{j=1}^{L} w_j C_j(\boldsymbol{x}) \\ &\text{subject to} && x_i \in \{0,1\} \qquad (1 \le i \le n). \end{aligned}$$

Because $C_j(\boldsymbol{x}) \le \sum_{i \in I_j^+} x_i + \sum_{i \in I_j^-}(1 - x_i)$, we use the following formulation of MAX 3SAT.

$$\begin{aligned} &\text{Maximize} \quad \sum_{j=1}^{L} w_j z_j \\ &\text{subject to} \\ &\quad z_j \le C_j(\boldsymbol{x}) \qquad (1 \le j \le L), \\ &\quad z_j \le \sum_{i \in I_j^+} x_i + \sum_{i \in I_j^-} (1 - x_i) \\ &\qquad\qquad\qquad (1 \le j \le L), \\ &\quad 0 \le z_j \le 1 \qquad (1 \le j \le L), \\ &\quad x_i \in \{0,1\} \qquad (1 \le i \le n). \end{aligned}$$

In the rest of this paper we omit the ranges of $i$ and $j$ if they are clear from the context.

Next we introduce variables $y_i$ ($i = 0, 1, \ldots, n$), whose values are $+1$ or $-1$ and replace $x_i$ with $(1 + y_0 y_i)/2$. Thus MAX 3SAT can be written as:

$$\begin{aligned} &\text{Maximize} \quad \sum_{j=1}^{L} w_j z_j \\ &\text{subject to} \quad z_j \le C'_j(\boldsymbol{y}), \\ &\qquad z_j \le \sum_{i \in I_j^+} \frac{1 + y_0 y_i}{2} + \sum_{i \in I_j^-} \frac{1 - y_0 y_i}{2}, \\ &\qquad 0 \le z_j \le 1, \\ &\qquad y_i \in \{-1, +1\}, \end{aligned}$$

where $C'_j(\boldsymbol{y}) = C_j((1 + y_0 y_1)/2, \ldots, (1 + y_0 y_n)/2)$.

Now we consider the case that $C_j$ has three literals, say, $x_i$, $x_j$ and $x_k$. In this case,

$$\begin{aligned} C'_j(\boldsymbol{y}) = \frac{1}{8}\big[&(1 + y_0 y_i) + (1 + y_0 y_j) + (1 + y_0 y_k) \\ &+ (1 - y_i y_j) + (1 - y_j y_k) + (1 - y_k y_i) \\ &+ (1 + y_0 y_i y_j y_k)\big]. \end{aligned}$$

Here, if there is a negative literal in those three literals, we replace the corresponding variable $y$ with $-y$. We denote by $c_j(\boldsymbol{y})$ and $d_j(\boldsymbol{y})$ the sum of the first six terms and the last term in $C'_j(\boldsymbol{y})$, respectively, that is, $C'_j(\boldsymbol{y}) = c_j(\boldsymbol{y}) + d_j(\boldsymbol{y})$ and $c_j(\boldsymbol{y})$ contains terms of degree at most 2. The next lemma bounds the value of $C'_j(\boldsymbol{y})$.

**Lemma 1** *For all $C_j$ containing three literals, $c_j(\boldsymbol{y}) \le C'_j(\boldsymbol{y}) \le (4/3)c_j(\boldsymbol{y})$.*

**Proof** The first inequality is trivial because $d_j(\boldsymbol{y}) \ge 0$. The second inequality comes as follows. Consider that clause $C_j$ has three literals. If $d_j(\boldsymbol{y}) = 0$, then $C'_j(\boldsymbol{y}) = c_j(\boldsymbol{y}) \ge 0$ and thus $(4/3)c_j(\boldsymbol{y}) \ge C'_j(\boldsymbol{y})$. We assume that $d_j(\boldsymbol{y}) \ne 0$. This means that $d_j(\boldsymbol{y}) = 1/4$ and $C'_j(\boldsymbol{y}) \ne 0$. Because $C'_j(\boldsymbol{y}) \in \{0, 1\}$, $C'_j(\boldsymbol{y})$ must equal 1. Thus, $c_j(\boldsymbol{y}) = 3/4$ and $C'_j(\boldsymbol{y}) \le (4/3)c_j(\boldsymbol{y})$. □

We define $c_j(\boldsymbol{y}) = C'_j(\boldsymbol{y})$ for all the clauses $C_j$ which have less than three literals. Thus we can reformulate MAX 3SAT as follows:

$$\begin{aligned} &\text{Maximize} \quad \sum_{j=1}^{L} w_j z_j \\ &\text{subject to} \\ &\quad z_j \le c_j(\boldsymbol{y}) \quad \text{if } |C_j| \le 2, \\ &\quad z_j \le \frac{4}{3} c_j(\boldsymbol{y}) \quad \text{if } |C_j| = 3, \\ &\quad z_j \le \sum_{i \in I_j^+} \frac{1 + y_0 y_i}{2} + \sum_{i \in I_j^-} \frac{1 - y_0 y_i}{2}, \\ &\quad 0 \le z_j \le 1, \\ &\quad y_i \in \{-1, +1\}. \end{aligned} \tag{1}$$

### 3.2 Reducing to a Semidefinite Programming Problem

In the problem (1), $c_j(\boldsymbol{y})$ contains the constant terms and the product terms of two variables. This fact enables us to relax the problem by using the following replacements:

(1) replace variables $y_i$ with vectors $\boldsymbol{v}_i$ whose norms are 1,

(2) replace the product $y_i y_j$ with the inner product of the corresponding vectors $\boldsymbol{v}_i \cdot \boldsymbol{v}_j$,

(3) replace the inner product $\boldsymbol{v}_i \cdot \boldsymbol{v}_j$ with a new variable $y_{ij}$.

Let $V$ be the matrix $(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ and let $V^T$ be its transpose. Then the $(n+1) \times (n+1)$ matrix $Y = (y_{ij})$ equals $V^T V$, which is symmetric and positive semidefinite. Thus we denote by $c_j(Y)$ a relaxed version of $c_j(\boldsymbol{y})$ and we have the maximization problem:

$$\begin{aligned}
&\text{Maximize} \quad \sum_{j=1}^{L} w_j z_j \\
&\text{subject to} \\
&\quad z_j \le c_j(Y) \quad \text{if } |C_j| \le 2, \\
&\quad z_j \le \frac{4}{3} c_j(Y) \quad \text{if } |C_j| = 3, \\
&\quad z_j \le \sum_{i \in I_j^+} \frac{1 + y_{0i}}{2} + \sum_{i \in I_j^-} \frac{1 - y_{0i}}{2}, \\
&\quad 0 \le z_j \le 1, \\
&\quad y_{ii} = 1, \\
&\quad \text{matrix } Y \text{ is symmetric and positive semidefinite.}
\end{aligned} \tag{2}$$

An optimization problem which is represented by a linear system of the entries of a symmetric and positive semidefinite matrix is called a "semidefinite programming problem." It is well-known that a semidefinite programming problem can be solved in polynomial time with respect to the input size and $\log(1/\epsilon)$ within any additive error $\epsilon$[1]. See Goemans and Williamson[4] for details and references.

For (2) we consider the matrix

$$\tilde{Y} = \begin{pmatrix} Y & O \\ O & \mathrm{diag}(z_1, z_2, \ldots, z_L) \end{pmatrix},$$

where $\mathrm{diag}(z_1, z_2, \ldots, z_L)$ is a diagonal matrix whose diagonal entries are $z_1, z_2, \ldots, z_L$. Then the objective function and all constraints in (2) are linear combinations of the entries of $\tilde{Y}$, and $\tilde{Y}$ is symmetric and positive semidefinite. Thus, (2) is a semidefinite programming, and hence we can find its solution $(Y^*, \boldsymbol{z}^*)$ (with a negligibly small error) in polynomial time.

### 3.3 Finding Approximate Solutions to MAX 3SAT

We use the following scheme, which is used in Goemans and Williamson[4], to obtain an approximate assignment $\boldsymbol{x}^S$ of MAX 3SAT:

First we obtain the vectors $\boldsymbol{v}_i^*$ by Cholesky decomposition of $Y^*$. Then we take the random vector $\boldsymbol{r}$ whose norm is 1 and let $y_i^S = \mathrm{sign}(\boldsymbol{r} \cdot \boldsymbol{v}_i^*)$. $x_i^S$ is set to $(1 + y_0^S y_i^S)/2$.

## 4. Analysis of the Approximation Ratio

In this section, we show the weight of the approximate assignment $\boldsymbol{x}^S$.

**Lemma 2** *The weight of $\boldsymbol{x}^S$ is bounded as follows.*

$$\sum_{j=1}^{L} w_j C_j(\boldsymbol{x}^S) \ge \alpha \sum_{|C_j| \le 2} w_j z_j^* + \frac{3\alpha}{4} \sum_{|C_j| = 3} w_j z_j^*,$$

*where* $\alpha = \min_{0 \le \theta \le \pi} (\theta/\pi)/[(1 - \cos\theta)/2] = 0.87856\ldots$.

To prove this lemma, we use the next lemma[4]. Proof is given so that this paper may be self-contained.

**Lemma 3** *For $\alpha$ in Lemma 2,*

$$E[(1 - y_i y_j)/2] \ge \alpha(1 - y_{ij}^*)/2,$$
$$E[(1 + y_i y_j)/2] \ge \alpha(1 + y_{ij}^*)/2.$$

**Proof** We can prove both inequalities in the same manner, so here we prove only the first one. Because the value of $(1 - y_i y_j)/2$ is either 0 or 1,

$$\begin{aligned}
&E[(1 - y_i y_j)/2] \\
&\quad = \Pr\{y_i \ne y_j\} \\
&\quad = \Pr\{\mathrm{sign}(\boldsymbol{r} \cdot \boldsymbol{v}_i^*) \ne \mathrm{sign}(\boldsymbol{r} \cdot \boldsymbol{v}_j^*)\} \\
&\quad = 2\Pr\{\boldsymbol{r} \cdot \boldsymbol{v}_i^* > 0 \text{ and } \boldsymbol{r} \cdot \boldsymbol{v}_j^* < 0\} \\
&\qquad \text{(by symmetry).}
\end{aligned}$$

We denote by $\theta_{ij}$ the angle between $\boldsymbol{v}_i^*$ and $\boldsymbol{v}_j^*$. The set $\{\boldsymbol{r} : \boldsymbol{r} \cdot \boldsymbol{v}_i^* > 0 \text{ and } \boldsymbol{r} \cdot \boldsymbol{v}_j^* < 0\}$ is a spherical lune with angle $\theta_{ij}$, whose measure equals $\theta_{ij}/(2\pi)$ times the measure of the full sphere. Thus

$$\begin{aligned}
&E[(1 - y_i y_j)/2] \\
&\quad = 2\Pr\{\boldsymbol{r} \cdot \boldsymbol{v}_i^* > 0 \text{ and } \boldsymbol{r} \cdot \boldsymbol{v}_j^* < 0\} \\
&\quad = 2\theta_{ij}/(2\pi) \\
&\quad = \theta_{ij}/\pi.
\end{aligned}$$

From the definition of $\alpha$, for any $\theta$ $(0 \le \theta \le \pi)$

$\theta/\pi \geq \alpha(1-\cos\theta)/2$. Thus

$$\begin{aligned} E[(1-y_iy_j)/2] &= \theta_{ij}/\pi \\ &\geq \alpha(1-\cos\theta_{ij})/2 \\ &= \alpha(1-y^*_{ij})/2. \end{aligned}$$ □

**Proof (of Lemma 2)** From Lemma 3, $E[c_j(\boldsymbol{y})] \geq \alpha c_j(Y^*)$. Thus for clauses $C_j$ which has at most two literals

$$E[C_j(\boldsymbol{x})] = E[c_j(\boldsymbol{y})] \geq \alpha c_j(Y^*) \geq \alpha z^*_j.$$

On the other hand, for the clauses $C_j$ which has three literals,

$$E[C_j(\boldsymbol{x})] \geq E[c_j(\boldsymbol{y})] \geq \alpha c_j(Y^*) \geq 3\alpha z^*_j/4.$$

Combining these inequalities,

$$\begin{aligned} &\sum_{j=1}^{L} w_j C_j(\boldsymbol{x}^S) \\ &\geq E\left[\sum_{j=1}^{L} w_j C_j(\boldsymbol{x})\right] \\ &= \sum_{j=1}^{L} w_j E[C_j(\boldsymbol{x})] \\ &= \sum_{|C_j|\leq 2} w_j E[C_j(\boldsymbol{x})] + \sum_{|C_j|=3} w_j E[C_j(\boldsymbol{x})] \\ &\geq \sum_{|C_j|\leq 2} w_j \alpha z^*_j + \sum_{|C_j|=3} w_j \cdot 3\alpha z^*_j/4 \\ &= \alpha \sum_{|C_j|\leq 2} w_j z^*_j + \frac{3\alpha}{4} \sum_{|C_j|=3} w_j z^*_j. \end{aligned}$$ □

## 5. 0.769-Approximation Algorithm for MAX 3SAT

In this section, we consider the approximation algorithm $A$ which selects one of three assignments $\boldsymbol{x}^J$, $\boldsymbol{x}^L$ and $\boldsymbol{x}^S$ with some probabilities. $\boldsymbol{x}^S$ is obtained by using the rounding scheme described in Section 3. $\boldsymbol{x}^J$ and $\boldsymbol{x}^L$ are obtained by the random assignment method[9], that is, each variable $x_i$ is independently set to 1 with probability $p_i$.

For $\boldsymbol{x}^J$, we let $p_i = 1/2$ for all $i$. For $\boldsymbol{x}^L$, we let $p_i = (1+y^*_{0i})/2$. Thus for the weights of $\boldsymbol{x}^J$ and $\boldsymbol{x}^L$, we can show the following lemmas.

**Lemma 4** *The weight of $\boldsymbol{x}^J$ is bounded as follows.*

$$\sum_{j=1}^{L} w_j C_j(\boldsymbol{x}^J) \geq \sum_{k=1}^{3}\left[\left(1-\frac{1}{2^k}\right)\sum_{|C_j|=k} w_j\right].$$

**Proof** Because we consider that $p_i = 1/2$ for all $i$, the probability that the clause $C_j$ is satisfied is $1 - 1/2^{|C_j|}$. Thus,

$$\begin{aligned} \sum_{j=1}^{L} w_j C_j(\boldsymbol{x}^J) &\geq E\left[\sum_{j=1}^{L} w_j C_j(\boldsymbol{x})\right] \\ &= \sum_{j=1}^{L} w_j E[C_j(\boldsymbol{x})] \\ &= \sum_{j=1}^{L} w_j \left(1-\frac{1}{2^{|C_j|}}\right) \\ &= \sum_{k=1}^{3}\left[\left(1-\frac{1}{2^k}\right)\sum_{|C_j|=k} w_j\right]. \end{aligned}$$ □

**Lemma 5** *The weight of $\boldsymbol{x}^L$ is bounded as follows.*

$$\sum_{j=1}^{L} w_j C_j(\boldsymbol{x}^L) \geq \sum_{k=1}^{3}\left[\beta_k \sum_{|C_j|=k} w_j z^*_j\right],$$

*where $\beta_k = 1-(1-1/k)^k$.*

**Proof** We can see that (2) is a version of LP relaxation of MAX SAT[5]. Thus, we can apply lemma 3.1 in Ref. 5) to $\boldsymbol{x}^L$ to obtain the bound of its weight:

$$\sum_{j=1}^{L} w_j C_j(\boldsymbol{x}^L) \geq \sum_{j=1}^{L} \beta_{|C_j|} w_j z^*_j,$$

where $\beta_k = 1-(1-1/k)^k$. Restricting MAX SAT to MAX 3SAT, we can obtain the lemma. □

Finally we show the approximation ratio of algorithm $A$. We assume that $A$ selects $\boldsymbol{x}^J$, $\boldsymbol{x}^L$ and $\boldsymbol{x}^S$ with the probabilities $p_J$, $p_L$, $p_S$, respectively.

**Theorem 1** *Algorithm $A$ is 0.769-approximation for appropriate $p_J$, $p_L$ and $p_S$.*

**Proof** Algorithm $A$ is $a$-approximation for the value of $a$ satisfying

$$\begin{aligned} \frac{1}{2}p_J + p_L + \alpha p_S &\geq a, \\ \frac{3}{4}p_J + \frac{3}{4}p_L + \alpha p_S &\geq a, \\ \frac{7}{8}p_J + \frac{19}{27}p_L + \frac{3\alpha}{4}p_S &\geq a. \end{aligned}$$

These inequalities express the expectations of $C_j(\boldsymbol{x})$ corresponding to the clauses with one, two and three literals, respectively. The upper bound of $a$, obtained by solving a linear programming problem, is 0.7694.... Thus algorithm $A$ is 0.769-approximation. □

## 6. Concluding Remarks

We have presented a randomized 0.769-

approximation algorithm for MAX 3SAT. The best previously known approximation algorithm for MAX 3SAT had 0.755 approximation ratio as a special case of the same bound for MAX SAT, so we have made a slight improvement by limiting MAX SAT to MAX 3SAT.

For MAX 3SAT, approximating within the ratio 38/39 is known to be NP-complete. The approximation ratio of our algorithm is much smaller than 38/39. A challenging problem is to narrow this gap.

**Acknowledgments** The authors would like to thank Michel Goemans for his comments on the first version of this paper, and anonymous referees for their valuable comments which improve the exposition of this paper.

## References

1) Alizadeh, F.: Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization, *SIAM J. Optimization*, Vol.5, No.1, pp.13–51 (1995).
2) Bellare, M., Goldreich, O. and Sudan, M.: Free Bits, PCPs and Non-Approximability – Towards Tight Results, Unpublished Manuscript.
3) Bellare, M., Goldreich, O. and Sudan, M.: Free Bits, PCPs and Non-Approximability – Towards Tight Results, TR95-024, ECCC, the Electronic Colloquium on Computational Complexity (1995).
4) Goemans, M.X. and Williamson, D.P.: .878-Approximation Algorithms for MAX CUT and MAX 2SAT, *Proc. 26th STOC*, pp.422–431 (1994).
5) Goemans, M.X. and Williamson, D.P.: New 3/4-Approximation Algorithms for the Maximum Satisfiability Problem, *SIAM J. Disc. Math.*, Vol.7, No.4, pp.656–666 (1994).
6) Johnson, D.S.: Approximation Algorithms for Combinatorial Problems, *J. Comput. Syst. Sci.*, Vol.9, pp.256–278 (1974).
7) Ono, T., Hirata, T. and Asano, T.: An Approximation Algorithm for MAX 3SAT, *Proc. 6th ISAAC*, pp.163–170 (1995).
8) Papadimitriou, C.H. and Yannakakis, M.: Optimization, Approximation, and Complexity Classes, *Proc. 20th STOC*, pp.229–234 (1988).
9) Raghavan, P. and Thompson, C.D.: Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs, *Combinatorica*, Vol.7, No.4, pp.365–374 (1987).
10) Yannakakis, M.: On the Approximation of Maximum Satisfiability, *Journal of Algorithms*, Vol.17, No.3, pp.475–502 (1994).

(Received May 22, 1966)
(Accepted July 4, 1996)

**Takao Ono** was born in 1970. He received the M.E. degree from Nagoya University in 1995. He is currently studying towards the Ph.D. degree in the Graduate School of Engineering in the same university. His research interests include approximation algorithms.

**Tomio Hirata** was born in 1949. He received the B.S., M.S. and Ph.D. in Computer Science, all from Tohoku University in 1976, 1978, and 1981, respectively. He is currently a Professor in Electronics Department, Nagoya University. From 1981 to 1986 he was an Assistant Professor of the Computer Science Department, Toyohashi University of Technology. His research interests include graph algorithms, data structures, and parallel algorithms.

**Takao Asano** was born in Miyagi Prefecture on January 7, 1949. He received the B.E., M.E. and D.E. degrees in Electrical Engineering and Communications from Tohoku University, Sendai, Japan, in 1972, 1974 and 1977, respectively. From 1977 to 1980, he was Research Associate of Tohoku University. From 1980 to 1985, he was Lecturer of Department of Mathematical Engineering of the University of Tokyo. From 1985 to 1992, he was Associate Professor of Sophia University. From 1992, he is Professor of Department of Information and System Engineering of Chuo University. His research interests include network algorithms and computational geometry.