HIROSHIMA MATH. J.
21 (1991), 351–383

An Algorithm for Constructing a Weight-Controlled Subset and Its Application to Graph Coloring Problem

Fumio KITAGAWA

(Received March 31, 1990)

Contents

Intoduction

An *existential problem of a weight-controlled subset*, which is abbreviated by a WSP, is a combinatorial problem proposed in [16]. This is specified by a 5-tuple

$$\langle U, \boldsymbol{S}, \omega\colon U \times \boldsymbol{S} \to Z_{+0}, a, b\colon \boldsymbol{S} \to Z_{+0} \rangle \tag{0.1}$$

of a finite set U, a collection $\boldsymbol{S}$ of subsets of U and non-negative integer valued functions ω, a, b, where the *weight function* satisfies

$$\begin{cases} \omega(u, s) > 0 & \text{if } u \in s, \\ \omega(u, s) = 0 & \text{otherwise,} \end{cases}$$

for any $u \in U$ and $s \in \boldsymbol{S}$. Then, the problem is to find a subset A of U satisfying

$$a(s) \leqq \Omega(A, s) = \sum_{u \in A} \omega(u, s) \leqq b(s) \quad \text{for any } s \in \boldsymbol{S}. \tag{0.2}$$

This problem is a general form of various combinatonial problems, e.g., the problem of timetables [6, 10, 12, 16], graph colorings [2, 20, 22, 23], network flows [8] or Latin squares [21]. It is difficult in general to check all $\Omega(A, s)$ when U and $\boldsymbol{S}$ are large, and several researches to solve the problems are done in these papers.

The purpose of this paper is to propose a new efficient algorithm to solve a WSP. To do this, for a given WSP specified by (0.1), we consider a subset $B \subset U$, called to be *bounded by b*, satisfying

(0.3) $$\Omega(B, s) = \sum_{u \in B} \omega(u, s) \leqq b(s) \quad \text{for any } s \in \boldsymbol{S},$$

where $\Omega(\emptyset, s) = 0$. If $\Omega(B, s) < a(s)$ for some $s \in \boldsymbol{S}$ and if there exists

(0.4) $$u \in \mathscr{V}(B, s) = \{u \in s - B \mid B \cup \{u\} \text{ is bounded by } b\},$$

then $\Omega(B, s) + 1 \leqq \Omega(B \cup \{u\}, s)$. Thus, starting from the bounded set $\emptyset$ by b and adding such an element u to B repeatedly, we may find A with (0.2). More precisely, we consider a function

$$\mathscr{A}: \mathbf{N} = \{0\} \cup (\bigcup_{k=1}^{p} (\{0\} \times Z_+^k)) \longrightarrow 2^U$$

$$(p = |U|, \text{ the number of elements in } U),$$

for the set Z_+ of all positive integers, satisfying the following conditions:
$(*)_0$ $\mathscr{A}(0) = \emptyset$ and $i_0 = 0$.
$(*)_k$ If $B = \mathscr{A}(i_0, \dots, i_k)$ is defined for some $k < p$ and $i_1, \dots, i_k \in Z_+$, then we define

$$\mathscr{A}(i_0, \dots, i_k, j) = \begin{cases} B \cup \{u_j\} & \text{if } \mathscr{S}(B) \neq \emptyset \text{ and } 1 \leqq j \leqq l, \\ B & \text{otherwise,} \end{cases}$$

where $\{u_1, \dots, u_l\}$ is the set with some order of all different elements of $\mathscr{V}(B, s)$ in (0.4) for some s in

$$\mathscr{S}(B) = \{s \in \boldsymbol{S} \mid \Omega(B, s) < a(s) \text{ and } \mathscr{V}(B, s) \neq \emptyset\}.$$

This function $\mathscr{A}$ gives us a tree T defined as follows:
$(**)_0$ Each node of T is an element of $\mathbf{N}$, and 0 is the root of T.
$(**)_k$ If $(i_0, \dots, i_k)$ is a node of T and $\mathscr{A}(i_0, \dots, i_k) \subsetneqq \mathscr{A}(i_0, \dots, i_k, j)$ for some $k < p$, $j \in Z_+$, then $(i_0, \dots, i_k, j)$ is a node of T and connected with node $(i_0, \dots, i_k)$.
$(**)$ Every node of T is given only by $(**)_0$ and a finite number of applications of $(**)_k$.

Now, the main result of this paper is stated as follows:

THEOREM. *Let T be the tree of function $\mathscr{A}$ of a given WSP specified by* (0.1) *and assume that there exists a solution of the WSP. Then there is a leaf node $(i_0, \dots, i_k)$ of T such that $\mathscr{A}(i_0, \dots, i_k)$ is a solution of the WSP.*

In other words, we can find a solution of a WSP by checking the first inequality in (0.2) for any leaf node $(i_0, \dots, i_k)$ in T if the WSP has a solution.

In §1, we discuss how to check the solvability of a WSP and to transform it simpler. In §2, we study the basic properties of the tree derived from a WSP, and prove the theorem. In §3, we give an algorithm for solving a WSP by a tree searching according to the theorem. In §4, we apply it to construct a solution of the 4-color problem; and in Appendix I & II, some FORTRAN programs derived from these are given.

1. An existential problem of a weight-controlled subset and some results

An existential problem of a wieght-controlled subset and its solution are defined in [16] as follows:

DEFINITION 1 (Specification of WSP). An *existential problem of a weight-controlled subset* is specified by a 5-tuple $\langle U, \boldsymbol{S}, \omega, a, b \rangle$ satisfying the following conditions:

(1) U is a finite set which is called the *universe*,

(2) $\boldsymbol{S}$ is a collection of subsets of U, which is called the *condition set*,

(3) ω is a function from the set $\{(u, s) | u \in s\}$ to the set of all positive integers, which is called the *weight function*,

(4) a is a function from $\boldsymbol{S}$ to the set of all non-negative integers, which is called the *lower bound function*,

(5) b is a function from $\boldsymbol{S}$ to the set of all non-negative integers, which is called the *upper bound function*,

DEFINITION 2 (Solution of WSP). A *solution* of WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$ is a subset A of U satisfying the condition:

$$a(s) \leqq \Omega(A \cap s, s) \leqq b(s) \quad \text{for any } s \text{ in } \boldsymbol{S},$$

where

$$\Omega(U', s) = \sum_{u \in U'} \omega(u, s)$$

for a subset U' of U.

By the definition of Ω, we have easily the following properties:

$$\Omega(U' \cap U'', s) = \Omega(U', s) - \Omega(U' - U'', s), \tag{1.1}$$

$$\Omega(U' \cup U'', s) = \Omega(U', s) + \Omega(U'', s) \quad \text{if } U' \cap U'' = \emptyset \tag{1.2}$$

for two subsets U' and U'' of U.

In the followings, we denote Z_{+0} for the set of all non-negative integers, Z_+ for the set of all positive integers and $\alpha(P)$ for the set of all solutions of a WSP P.

In order to expand a partial function ω on $U \times \boldsymbol{S}$ to a total function, we define $\tilde{\omega}$ as follows:

$$\tilde{\omega}(u, s) = \begin{cases} \omega(u, s) & \text{if } u \text{ in } s, \\ 0 & \text{otherwise.} \end{cases}$$

We use ω in introduction instead of $\tilde{\omega}$.

Figure 1 gives an example of a WSP with $U = \{u_1, u_2, \ldots, u_8\}$, $\boldsymbol{S} = \{s_1, s_2, \ldots, s_5\}$. Each value of $\omega(u_i, s_j)$ is appeared at the cross point of row u_i and column s_j. Values $a(s_i)$ and $b(s_i)$ are represented at the column s_i of row a and row b respectively. Column A indicates a solution $\{u_1, u_4, u_6, u_7\}$ of the WSP.

$\tilde{\omega}$ $\boldsymbol{S}$ / U	s_1	s_2	s_3	s_4	s_5	A
u_1	1	0	2	0	4	1
u_2	0	3	0	0	3	0
u_3	0	0	1	2	0	0
u_4	2	3	0	1	3	1
u_5	0	1	2	0	0	0
u_6	1	0	2	0	0	1
u_7	0	2	0	3	1	1
u_8	2	1	0	2	0	0
a	4	5	4	3	7	
b	5	6	5	4	9	

Figure 1 An example of a WSP

If the size of U and/or $\boldsymbol{S}$ of a WSP is larger, it is more difficult to solve it. Therefore it is important to reduce U and/or $\boldsymbol{S}$ without any change of solvability.

DEFINITION 3 (Inconsistency). A WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$ is said to be *inconsistent* if there is an element s in $\boldsymbol{S}'$ such that $a'(s) > b'(s)$ on a WSP $\langle U, \boldsymbol{S}', \omega', a', b'\rangle$, where

$$\boldsymbol{S}' = \boldsymbol{S} \cup \{s'\},$$

$$\omega'(u, s) = \begin{cases} \omega(u, s) & \text{if } s \in \boldsymbol{S} \text{ and } u \in s, \\ \max\{\omega(u, s_i) \mid u \in s_i,\ i = 1, 2, \ldots, k\} & \text{if } s = s' \text{ and } u \in s', \end{cases}$$

$$a'(s) = \begin{cases} a(s) & \text{if } s \in \boldsymbol{S}, \\ \max(\{a(s_i) - \Omega(s_i - s', s_i) \mid i = 1, 2, \ldots, k\} \cup \{0\}) & \text{if } s = s', \end{cases}$$

$$b'(s) = \begin{cases} b(s) & \text{if } s \in \boldsymbol{S} \\ \min(\{b(s_i) + \sum_{u \in s'} (\omega'(u, s') - \omega(u, s_i)) \mid i = 1, 2, \ldots, k\} \\ \qquad \cup \{\Omega(s', s')\}) & \text{if } s = s', \end{cases}$$

and

$$s' = \bigcap_{i=1}^{k} s_i .$$

Kitagawa [16] proved that if a WSP is inconsistent then there is no solution. We give an example of an inconsistent WSP in Figure 2. It is difficult to check that the problem has no solution without checking each subset of U. But we can prove that by adding a new column $s_1 \cap s_2$.

$\tilde{\omega}$	$\boldsymbol{S}$	s_1	s_2	$s_1 \cap s_2$
U				
u_1		2	2	2
u_2		3	1	3
u_3		0	1	0
u_4		2	2	2
a		5	1	5
b		5	2	4

Figure 2 An example of an inconsistent WSP

Kitagawa [16], therefore, proposed a conjecture "*if a WSP has no solution then it is inconsistent*". But we find a counter-example of the conjecture shown in Figure 3. In this example, let be given a WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$ with $\boldsymbol{S} = \{s_1, s_2\}$ and $U = \{u_1, u_2\}$. If a new WSP $\langle U, \boldsymbol{S}', \omega', a', b'\rangle$ is created as an extension of $\boldsymbol{S}'$ such that

$$\boldsymbol{S}' = \boldsymbol{S} \cup \{s'\},$$

$$a'(s) = \begin{cases} a(s) & \text{if } s \in \boldsymbol{S}, \\ 1 & \text{if } s = s', \end{cases}$$

$$b'(s) = \begin{cases} b(s) & \text{if } s \in \boldsymbol{S}, \\ 1 & \text{if } s \in s', \end{cases}$$

$$\omega'(u, s) = \begin{cases} \omega(u, s) & \text{if } u \in s \text{ and } s \in \boldsymbol{S}, \\ 1 & \text{if } s = s' \text{ and } u = u_1, \\ 0 & \text{is } s = s' \text{ and } u = u_2, \end{cases}$$

then the new WSP has no solution, nevertheless $a'(s') = b'(s')$.

$\tilde{\omega}$	$\boldsymbol{S}$	s_1	s_2	$s' = s_1 \cap s_2$
U				
u_1		1	1	1
u_2		0	2	0
a		1	2	1
b		1	2	1

Figure 3 A counter example of the conjecture in [16]

As similar as expansion of WSP by intersection, we can expand a WSP by union as follows.

DEFINITION 4 (Expansion by union). For $P = \text{WSP}\langle U, \boldsymbol{S}, \omega, a, b\rangle$, the expansion of P by union is a $\text{WSP}\langle U, \tilde{\boldsymbol{S}}, \omega', a', b'\rangle$ which is defined as follows:

$$\tilde{\boldsymbol{S}} = \boldsymbol{S} \cup \{s_{\boldsymbol{S}}\},$$

$$\omega'(u, s) = \begin{cases} \omega(u, s) & \text{if } s \in \boldsymbol{S} \text{ and } u \in s, \\ \displaystyle\sum_{\{s'' \in \boldsymbol{S} \mid u \in s''\}} \omega(u, s'') & \text{if } s = s_{\boldsymbol{S}}, \end{cases}$$

$$a'(s) = \begin{cases} a(s) & \text{if } s \in \boldsymbol{S}, \\ \min\{\sum_{s \in \boldsymbol{S}} \Omega(s \cap A, s) \mid A \in \alpha(\underset{\sim}{P})\} & \text{if } s = s_{\boldsymbol{S}}, \end{cases}$$

$$b'(s) = \begin{cases} b(s) & \text{if } s \in \boldsymbol{S}, \\ \max\{\sum_{s \in \boldsymbol{S}} \Omega(s \cap A, s) \mid A \in \alpha(\tilde{P})\} & \text{if } s = s_{\boldsymbol{S}}, \end{cases}$$

where

$$s_{\boldsymbol{S}} = \bigcup_{s \in \boldsymbol{S}} s,$$

$$\tilde{P} = \text{WSP}\langle U, \boldsymbol{S}, \omega, a_0, b\rangle,$$

$$\underset{\sim}{P} = \text{WSP}\langle U, \boldsymbol{S}, \omega, a, b_0\rangle,$$

$$a_0(s) = 0 \text{ for any } s \text{ in } \boldsymbol{S},$$

$$b_0(s) = \max\{\Omega(s'', s'') \mid s'' \in \boldsymbol{S}\} \text{ for any } s \text{ in } \boldsymbol{S}.$$

THEOREM 1. *If A is a solution of WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$, then it is also a solution of the expansion WSP $\langle U, \tilde{\boldsymbol{S}}, \omega', a', b'\rangle$ by union.*

PROOF. Let A be a solution of WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$. A is also a solution of both $\underset{\sim}{P}$ and $\tilde{P}$. Then, $\Omega(s_{\boldsymbol{S}} \cap A, s_{\boldsymbol{S}}) \geqq a'(s_{\boldsymbol{S}})$ can be proved as follows.

$$
\begin{aligned}
&\Omega(s_{\boldsymbol{S}} \cap A, s_{\boldsymbol{S}}) \\
&= \sum_{u \in s_{\boldsymbol{S}} \cap A} \omega'(u, s_{\boldsymbol{S}}) \\
&= \sum_{u \in s_{\boldsymbol{S}} \cap A} \sum_{\{s' \in \boldsymbol{S} \mid u \in s'\}} \omega(u, s') && \text{by the definition of } \omega'(u, s_{\boldsymbol{S}}) \\
&= \sum_{s' \in \boldsymbol{S}} \sum_{u \in s' \cap A} \omega(u, s') && \text{by } s_{\boldsymbol{S}} = \textstyle\bigcup_{s \in \boldsymbol{S}} s \\
&\geqq \min\{\sum_{s \in \boldsymbol{S}} \Omega(s \cap A', s) \mid A' \in \alpha(\underset{\sim}{P})\} && \text{by } A \in \alpha(\underset{\sim}{P}) \\
&= a'(s_{\boldsymbol{S}}) && \text{by the definitin of } a'(s_{\boldsymbol{S}}).
\end{aligned}
$$

Similarly, $\Omega(s_{\boldsymbol{S}} \cap A, s_{\boldsymbol{S}}) \leqq b'(s_{\boldsymbol{S}})$ can be proved as follows.

$$
\begin{aligned}
&\Omega(s_{\boldsymbol{S}} \cap A, s_{\boldsymbol{S}}) \\
&= \sum_{s' \in \boldsymbol{S}} \sum_{u \in s' \cap A} \omega(u, s') \\
&\leqq \max\{\sum_{s \in \boldsymbol{S}} \Omega(s \cap A', s) \mid A' \in \alpha(\tilde{P})\} && \text{by } A \in \alpha(\tilde{P}) \\
&= b'(s_{\boldsymbol{S}}) && \text{by the definition of } b'(s_{\boldsymbol{S}}).
\end{aligned}
$$

Therefore

$$a'(s_{\boldsymbol{S}}) \leqq \Omega(s_{\boldsymbol{S}} \cap A, s_{\boldsymbol{S}}) \leqq b'(s_{\boldsymbol{S}}).$$

(Q.E.D.)

By Theorem 1, we obtain the following corollary.

COROLLARY 1. *In the expansion WSP $\langle U, \tilde{\boldsymbol{S}}, \omega', a', b'\rangle$ of a WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$, if the subset $s_{\boldsymbol{S}}$ in $\boldsymbol{S}$ satisfies*

$$a'(s_{\boldsymbol{S}}) > b'(s_{\boldsymbol{S}}),$$

then the WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$ has no solution.

By using Corollary 1, we can know that some WSPs have no solution.

Figure 4 gives an application of Corollary 1 to the WSP in Figure 3. In Figure 4, since $A_1 = \{u_1\}$, $A_2 = \{u_2\}$ and $A_3 = \{u_1, u_2\}$, we have $\alpha(\tilde{P}) = \{A_1, A_2\}$ and $\alpha(\underset{\sim}{P}) = \{A_3\}$. Then the value of $\sum_{s \in S} \Omega(s \cap A_i, s)$ is 2, 2, 4 with respect of each i in $\{1, 2, 3\}$. Then we have $a'(s_S) = 4$, $b'(s_S) = \max\{2, 2\} = 2$. Hence the WSP has no solution.

$\tilde{\omega}$ S	s_1	s_2	s_S
U			
u_1	1	1	2
u_2	0	2	2
a'	1	2	4
b'	1	2	2

Figure 4 An applicatin of Corollary 1

But the converse of Corollary 1 is not true except the case that S has only one element. Then, by Corollary 1, we cannot check whether a WSP has a solution or not. Figure 5 shows an example so that the converse of Corollary 1 is not true. In Figure 5, since $A_1 = \{u_1\}$, $A_2 = \{u_2\}$, $A_3 = \{u_3\}$, $A_4 = \{u_1, u_2\}$, $A_5 = \{u_1, u_3\}$, $A_6 = \{u_2, u_3\}$ and $A_7 = \{u_1, u_2, u_3\}$, we have $\alpha(\tilde{P}) = \{A_1, A_2, A_3, A_4\}$ and $\alpha(\underset{\sim}{P}) = \{A_5, A_6, A_7\}$. Then the value of $\sum_{s \in S} \Omega(s \cap A_i, s)$ is 4, 2, 3, 6, 7, 5, 9 with respect of each i in $\{1, 2, \ldots, 7\}$. Then we have $a'(s_S) = \min\{7, 5, 9\} = 5$ and $b'(s_S) = \max\{4, 2, 3, 6\} = 6$. Hence theWSP $\langle U, S, \omega, a, b\rangle$ has no solution, nevertheless $a'(s_S) < b'(s_S)$.

$\tilde{\omega}$ S	s_1	s_2	s_S
U			
u_1	3	1	4
u_2	1	1	2
u_3	0	3	3
a'	1	3	5
b'	4	3	6

Figure 5 An inapplicable example to Corollary 1

We have never find any simple (like Corollary 1), necessary and sufficient condition to check the solvability of WSPs. Then we will change our focus on special cases of WSPs such that weight function ω has the constant value 1. We will call it *binary WSP or BWSP.* Actually, we can prove that each WSP can be transformed to a BWSP without loss of generality. In order to prove Theoremm 2, four lemmas will be proved in advance.

LEMMA 1. *For an element s in $\boldsymbol{S}$ and a non-empty subset s' of U satisfying two inequalities:*

$$a(s) \leqq \Omega(s, s) \tag{1.3}$$

and

$$\Omega(s - s', s) < a(s), \tag{1.4}$$

in a WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$, there is a subset s'' of s' such that

$$a(s) - \Omega(s - s', s) \leqq \Omega(s'' \cap s, s).$$

PROOF. It is easy to prove Lemma 1 as follows:

$$\begin{aligned} 0 &< a(s) - \Omega(s - s', s) && \text{by (1.4)} \\ &\leqq \Omega(s, s) - \Omega(s - s', s) && \text{by (1.3)} \\ &= \Omega(s' \cap s, s) && \text{by (1.1).} \end{aligned}$$

(Q.E.D.)

Since a BWSP can be specified as a 4-tuple $\langle U, \boldsymbol{S}, a, b\rangle$, we denote it by BWSP $\langle U, \boldsymbol{S}, a, b\rangle$.

Then, a *solution* A of a BWSP $\langle U, \boldsymbol{S}, a, b\rangle$ can be specified as a subset of U satisfying the following condition:

$$a(s) \leqq |A \cap s| \leqq b(s) \quad \text{for any } s \text{ in } \boldsymbol{S}.$$

LEMMA 2. *A BWSP $\langle U, \boldsymbol{S}', a', b'\rangle$ can be constructed from a WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$.*

PROOF. If $\omega(u, s) = 1$ for any u in s and any s in $\boldsymbol{S}$, then the 4-tuple $\langle U, \boldsymbol{S}, a, b\rangle$ is a BWSP.

If $\omega(u, s) > 1$ for some u in U and some s in $\boldsymbol{S}$, then let $\boldsymbol{S}'$, a' and b' be given as follows:

$$\boldsymbol{S}' = 2^U,$$

$$a'(s') = \max\{p(s', s) | s \in \boldsymbol{S}\} \quad \text{for any } s' \text{ in } \boldsymbol{S}', \tag{1.5}$$

$$b'(s') = \min\{q(s', s) | s \in \boldsymbol{S}\} \quad \text{for any } s' \text{ in } \boldsymbol{S}', \tag{1.6}$$

where

$$p(s', s) = \begin{cases} |s'| + 1 \quad \text{if } \Omega(s, s) < a(s), & (1.7) \\ \min\{|s''|: a(s) - \Omega(s - s', s) \leqq \Omega(s'' \cap s, s),\ s'' \subset s'\} & \\ \qquad \text{if } \Omega(s, s) \geqq a(s) \text{ and } \Omega(s - s', s) < a(s), & (1.8) \\ 0 \quad \text{if } \Omega(s, s) \geqq a(s) \text{ and } \Omega(s - s', s) \geqq a(s), & (1.9) \end{cases}$$

$$q(s', s) = \begin{cases} \max\{|s''|: b(s) \geqq \Omega(s'' \cap s, s),\ s'' \subset s'\} \text{ if } \Omega(s \cap s', s) > b(s), & (1.10) \\ |s'| \quad \text{if } \Omega(s \cap s', s) \leqq b(s), & (1.11) \end{cases}$$

and, $|s'|$ and $|s''|$ are the number of elements of s' and s'' respectively. It is obvious that the 4-tuple $\langle U, \boldsymbol{S}', a', b' \rangle$ is a BWSP. (Q.E.D.)

LEMMA 3. *Let 4-tuple $\langle U, \boldsymbol{S}', \boldsymbol{a}', \boldsymbol{b}' \rangle$ be the BWSP derived from a WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$. Then, there is an element s' in $\boldsymbol{S}'$ such that*

$$a'(s') = |s'| + 1,$$

iff there is an element s in $\boldsymbol{S}$ such that

$$\Omega(s, s) < a(s).$$

PROOF. (Necessity) If there is an element s' in $\boldsymbol{S}'$ such that $a'(s') = |s| + 1$, then there is an element s in $\boldsymbol{S}$ such that $p(s', s) = |s'| + 1$ by (1.5). Thus, there is an element s in $\boldsymbol{S}$ such that $\Omega(s, s) < a(s)$ by (1.7).

(Sufficiency) If there is an element s in $\boldsymbol{S}$ such that $\Omega(s, s) < a(s)$, then there is an element s' in $\boldsymbol{S}'$ such that $p(s', s) = |s'| + 1$ by (1.7). Any s in $\boldsymbol{S}$ such that $\Omega(s, s) \geqq a(s)$ satisfies $p(s', s) \leqq |s'|$ by (1.8) and (1.9). Hence

$$a'(s') = \max\{p(s', s) | s \in \boldsymbol{S}\} = |s'| + 1. \qquad \text{(Q.E.D.)}$$

LEMMA 4. *Let 4-tuple $\langle U, \boldsymbol{S}', a', b' \rangle$ be the BWSP derived from a WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$. Then any element s' in $\boldsymbol{S}'$ satisfies*

$$b'(s') \leqq |s'|.$$

PROOF. It is easy to prove the lemma as follows.

$$\begin{aligned} b'(s') &\leqq q(s', s) && \text{by (1.6)} \\ &\leqq |s'| && \text{by (1.10) and (1.11).} \end{aligned}$$

(Q.E.D.)

THEOREM 2. *Let P be a WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$. There exists a P'*

$= BWSP\langle U, \boldsymbol{S}, a', b'\rangle$ *such that*

P has a solution A iff P′ has a solution A.

PROOF. Let P' be the BWSP $\langle U, \boldsymbol{S}', a', b'\rangle$ derived from a WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$ P.

(Necessity) Let A be a solution of P. We will prove that A is a solution of P' in two cases (i) and (ii).

(i) For s' in $\boldsymbol{S}'$ satisfying $A \cap s' \neq \emptyset$, we will show that s' satisfies $a'(s') \leqq |A \cap s'| \leqq b'(s')$. Condition $a'(s') \leqq |A \cap s'|$ is proved in the following three cases (i.i), (i.ii) and (i.iii).

(i.i) Since P has a solution, there is no case such that $\Omega(s, s) < a(s)$. Then, there is no s' in $\boldsymbol{S}'$ such that $a'(s') = |s'| + 1$ by (1.7).

(i.ii) In the case that $\Omega(s, s) \geqq a(s)$ and $\Omega(s - s', s) < a(s)$, we have $p(s', s) = 0$ by (1.9).

(i.iii) In the case that $\Omega(s, s) \geqq a(s)$ and $\Omega(s - s', s) \geqq a(s)$, we have the following formulas.

$$\begin{aligned} a(s) &\leqq \Omega(A \cap s, s) && \text{by } A \in \alpha(P) \\ &= \Omega(A \cap s - A \cap s' \cap s, s) + \Omega(A \cap s' \cap s, s) && \text{by (1.2)} \\ &\leqq \Omega(s - s', s) + \Omega(A \cap s' \cap s, s). \end{aligned}$$

Since $A \cap s'$ satisfies the condition in (1.8), assign $A \cap s'$ to s'' in (1.8), and we have

$$p(s', s) \leqq |A \cap s'|.$$

From the above three cases, we finally obtain

$$a(s') \leqq |A \cap s'| \qquad \text{by (1.5)}.$$

Similar to the above discussion, we will show $|A \cap s| \leqq b'(s)$ in the following two cases (i.iv) and (i.v).

(i.iv) In the case that $\Omega(s \cap s', s) \leqq b(s)$, we have

$$q(s', s) = |s'| \qquad \text{by (1.11)}.$$

(i.v) In the case that $\Omega(s \cap s', s) > b(s)$, we have

$$\begin{aligned} b(s) &\geqq \Omega(A \cap s, s) && \text{by } A \in \alpha(P) \\ &= \Omega(A \cap s - A \cap s' \cap s, s) + \Omega(A \cap s' \cap s, s) && \text{by (1.2)} \\ &\geqq \Omega(A \cap s' \cap s, s). \end{aligned}$$

Since $A \cap s'$ satisfies the condition in (1.10), assign $A \cap s'$ to s'' in (1.10), and we have

$$q(s', s) \geqq |A \cap s'|.$$

From the above two cases, we obtain

$$b'(s') \geqq |A \cap s'| \qquad \text{by (1.6).}$$

Hence

$$a'(s') \leqq |A \cap s'| \leqq b'(s').$$

(ii) For s' in S' satisfying $A \cap s' = \emptyset$, we have

$$\begin{aligned} \Omega(s - s', s) &\geqq \Omega((s - s') \cap A, s) \\ &= \Omega(s \cap A, s) \qquad \text{by } A \cap s' = \emptyset \\ &\geqq a(s) \qquad \text{by } A \in \alpha(P), \end{aligned}$$

for any s in S. Then, we have

$$a'(s) = 0 \qquad \text{for any } s \text{ in } S.$$

On the other hand, we have

$$\begin{aligned} |A \cap s'| &= 0 \qquad \text{by } A \cap s' = \emptyset \\ &\leqq b'(s) \qquad \text{by the definition of } b'. \end{aligned}$$

Hence

$$a'(s') \leqq |A \cap s'| \leqq b'(s).$$

This completes the proof of the necessary condition.

(Sufficiency) Let A be a solution of P'. We will prove that A is a solution of P in two cases (i) and (ii).

(i) In the case that $A = \emptyset$, we have $a'(s') = 0$ for any s' in S'. The value of $p(s', s)$ in (1.8) is a positive integer by Lemma 1. The value of $p(s', s)$ in (1.7) is also a positive integer by the definition. Therefore $a'(s')$ is given by only (1.9), that is,

$$\Omega(s - s', s) \geqq a(s) \qquad \text{for any } s' \text{ in } S' \text{ and } s \text{ in } S.$$

Since $S' \supset S$, let s' be s, and we have

$$a(s) = 0 \qquad \text{for any } s \text{ in } S.$$

As the range of upper bound function b is non-negative integers, we have

$$a(s) \leqq \Omega(A \cap s, s) = 0 \leqq b(s) \qquad \text{for any } s \text{ in } S.$$

(ii) In the case that $A \neq \emptyset$, we suppose that A is not a solution of

P. There is an s' in 2^U such that $s' = A$. Then, we have

$$a'(A) \leqq |A| \leqq b'(A).$$

Hence

$$b'(A) = |A| \qquad \text{by lemma 4.} \tag{1.12}$$

On the other hand, if there exist three subsets s, s' of U and s'' of s' such that $\Omega(s \cap s', s) > b(s)$ and $\Omega(s \cap s'', s) \leqq b(s)$, then $s'' \subsetneqq s'$. Because, if $s'' = s'$, then these two inequalities contradict each other. Thus, if there are s and s'' such that $\Omega(s \cap s', s) > b(s)$, then $b'(s') < |s'|$ by (1.6) and (1.10). Therefore, if there is an s' in $\boldsymbol{S}'$ such that $b'(s') = |s'|$, then $\Omega(s \cap s', s) \leqq b(s)$ for any s in $\boldsymbol{S}$ by (1.11). Thus we have

$$\Omega(A \cap s, s) \leqq b(s) \qquad \text{for any } s \text{ in } \boldsymbol{S} \quad \text{by (1.12)}.$$

This inequality means that A satisfies the boundary condition b in P. From the above assumption that A is not a solution of P, there is an element s'' in $\boldsymbol{S}$ which does not satisfy the boundary condition a, that is.

$$\Omega(A \cap s'', s'') < a(s''). \tag{1.13}$$

If $A = U$, we have $\Omega(s'', s'') < a(s'')$. Then, there is an element s' in $\boldsymbol{S}'$ such that $a'(s') = |s'| + 1$ by Lemma 3. This contradicts the assumption that P' has a solution A. Therefore $A \subsetneqq U$, that is, $U - A \supsetneqq \emptyset$. Let $s' = U - A$, and we have

$$\begin{aligned} \Omega(s'' - (U - A), s'') &= \Omega(A \cap s'', s'') && \text{by } s'' \subset U \text{ and } A \subset U \\ &< a(s'') && \text{by (1.13).} \end{aligned}$$

Then, $(U - A)$ satisfies the condition formula in (1.8). Then, we have

$$\begin{aligned} 0 &< p(U - A, s'') && \text{by Lemma 1} \\ &\leqq a'(U - A) && \text{by (1.5).} \end{aligned}$$

This contradicts the assumption that A is a solution of P'. This completes the proof of the sufficient condition. (Q.E.D.)

Now we will give a useful proposition for reducing a BWSP.

PROPOSITION 1. *Suppose that there exist s_1 and s_2 in $\boldsymbol{S}$ satisfying the following conditions:*

$$s_1 \supset s_2, \tag{1.14}$$

$$a(s_1) \leqq a(s_2), \tag{1.15}$$

$$b(s_1) - b(s_2) \geqq |s_1| - |s_2|, \tag{1.16}$$

in a BWSP $\langle U, \boldsymbol{S}, a, b\rangle$. *Then a solution of the BWSP* $\langle U, \boldsymbol{S}, a, b\rangle$ *is also a solution of BWSP* $\langle U, \boldsymbol{S} - \{s_1\}, a|_{\boldsymbol{S}-\{s_1\}}, b|_{\boldsymbol{S}-\{s_1\}}\rangle$, *and vice-versa.*

PROOF. (i) Let A be given a solution of the BWSP $\langle U, \boldsymbol{S}, a, b\rangle$, and we have

$$a(s) \leqq |A \cap s| \leqq b(s) \quad \text{for any } s \text{ in } \boldsymbol{S}.$$

Since $\boldsymbol{S} - \{s_1\}$ is a subset of $\boldsymbol{S}$, A is also a solution of the BWSP $\langle U, \boldsymbol{S} - \{s_1\}, s|_{\boldsymbol{S}-\{s_1\}}, b|_{\boldsymbol{S}-\{s_1\}}\rangle$.

(ii). Let A be a solution of the BWSP $\langle U, \boldsymbol{S} - \{s_1\}, a|_{\boldsymbol{S}-\{s_1\}}, b|_{\boldsymbol{S}-\{s_1\}}\rangle$, and we have

$$a(s) \leqq |A \cap s| \leqq b(s) \quad \text{for any } s \text{ in } \boldsymbol{S} - \{s_1\}. \tag{1.17}$$

On the other hand, since

$$\begin{aligned} a(s_1) &\leqq a(s_2) && \text{by (1.15)} \\ &\leqq |A \cap s_2| && \text{by (1.17)} \\ &\leqq |A \cap s_1| && \text{by (1.14)}, \end{aligned}$$

and

$$\begin{aligned} |A \cap s_1| &\leqq |A \cap s_2| + |A \cap (s_1 - s_2)| && \\ &\leqq b(s_2) + |A \cap (s_1 - s_2)| && \text{by (1.17)} \\ &\leqq b(s_2) + |s_1 - s_2| && \\ &= b(s_2) + |s_1| - |s_2| && \text{by (1.14)} \\ &\leqq b(s_1) && \text{by (1.16)}, \end{aligned}$$

then we have

$$a(s_1) \leqq |A \cap s_1| \leqq b(s_1) \quad \text{for any } s \text{ in } \boldsymbol{S}.$$

Thus A is also a solution of the BWSP $\langle U, \boldsymbol{S}, a, b\rangle$. (Q.E.D.)

Figure 6 gives an example of Proposition 1. In this example, s_1 and s_2 satisfy the conditions in Proposition 1, and then we can remove s_1 from the BWSP without change of solvability.

S	s_1	s_2	s_3		S	s_2	s_3
U					U		
u_1	1	1	0		u_1	1	0
u_2	1	1	1		u_2	1	1
u_3	1	0	1	$\Longrightarrow$	u_3	0	1
u_4	0	0	1		u_4	0	1
a	1	1	1		a	1	1
b	2	1	2		b	1	2

Figure 6 An example of Proposition 1

We have discussed how to check solvabilitiy of a WSP and to make it simpler. This proposition is very useful, because the number of all subsets of U is too large to be checked exhaustively when the size of U is large. On the other hand, our purposes are not only to check the solvability, but also to construct a solution. In the following sections, we will show an efficient method for constructing a solution of WSP.

2. A tree derived from function $\mathscr{A}$ on WSPs

In order to provide an algorithm for construcing a solution of WSP, we will define a function of a given WSP and investigate its properties. To define the function for a given WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$, we use the following notations:

$$\mathscr{V}(U', s) = \{u \in s - U' \mid \Omega((U' \cup \{u\}) \cap s, s) \leqq b(s) \quad \text{for any } s \in \boldsymbol{S}\},$$

$$\mathscr{S}(U') = \{s \in \boldsymbol{S} \mid \Omega(U' \cap s, s) < a(s) \text{ and } \mathscr{V}(U', s) \neq \emptyset\},$$

for $U' \subset U$.

DEFINITION 5 (Function $\mathscr{A}$ of a WSP). For a given WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$, we define a function inductively as follows:

$$\mathscr{A}: \{0\} \cup (\textstyle\bigcup_{k=1}^{p} (\{0\} \times Z_+^k)) \longrightarrow 2^U$$

$$(p = |U|, \text{ the number of elements in } U),$$

for the set Z_+ of all positive integers, satisfying the following conditions:
$(*)_0$ $\mathscr{A}(0) = \emptyset$ and $i_0 = 0$.
$(*)_k$ If $B = \mathscr{A}(i_0, \ldots, i_k)$ is defined for some $k < p$ and $i_1, \ldots, i_k \in Z_+$, then we define

$$\mathscr{A}(i_0, \dots, i_k, j) = \begin{cases} B \cup \{u_j\} & \text{if } \mathscr{S}(B) \neq \emptyset \text{ and } 1 \leqq j \leqq l, \\ B & \text{otherwise,} \end{cases}$$

where $\{u_1, \dots, u_l\}$ is the set with some order of all different elements of $\mathscr{V}(B, s)$ for some s in $\mathscr{S}(B)$.

This function $\mathscr{A}$ gives us the labeled tree $T(P)$ of a given WSP P.

DEFINITION 6 (Tree derived from function $\mathscr{A}$ on WSPs). For a given WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$ P, we will define a labeled tree $T(P)$ as follows:

$(**)_0$ Each node of $T(P)$ is an element of $\{0\} \cup (\bigcup_{k=1}^{|U|} (\{0\} \times Z_+^k))$, and 0 is the root of $T(P)$.
$(**)_k$ If $(i_0, \dots, i_k)$ is a node of $T(P)$ and $\mathscr{A}(i_0, \dots, i_k) \subsetneqq \mathscr{A}(i_0, \dots, i_k, j)$ for some $k < p$, $j \in Z_+$, then $(i_0, \dots, i_k, j)$ is a node of $T(P)$ and connected with node $(i_0, \dots, i_k)$.
$(**)$ Every node of $T(P)$ is given only by $(**)_0$ and a finite number of applications of $(**)_k$.
The label of a node $(i_0, \dots, i_k)$ of $T(P)$ is $\mathscr{A}(i_0, \dots, i_k)$.

From the definition of function $\mathscr{A}$, each node of the $T(P)$ of a given WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$ P satisfies a property:

$$\Omega(\mathscr{A}(i_0, \dots, i_k) \cap s, s) \leqq b(s)$$

for any s in $\boldsymbol{S}$ and any node $(i_0, \dots, i_k)$ of $T(P)$. We also have a property:

$$\sum_{s \in \boldsymbol{S}} \Omega(\mathscr{A}(i_0, \dots, i_k) \cap s, s) + 1 \leqq \sum_{s \in \boldsymbol{S}} \Omega(\mathscr{A}(i_0, \dots, i_k, j) \cap s, s)$$

for any two nodes $(i_0, \dots, i_k)$, $(i_0, \dots, i_k, j)$ of $T(P)$ such that $(i_0, \dots, i_k, j)$ is connected with $(i_0, \dots, i_k)$.

Now, we show a relationship between a solution of a WSP P and $T(P)$ in Theorem 3. In order to prove it, a lemma is proved in advance.

LEMMA 5. *Let A be a solution of a given WSP $\langle U, \boldsymbol{S}, \omega, a, b\rangle$. Then, any subset U' of A satisfies the following condition*:

$$J(U') \neq \emptyset \Longrightarrow (\forall s \in J(U'))(A \cap \mathscr{V}(U', s) \neq \emptyset),$$

where

$$J(U') = \{s \in \boldsymbol{S} \mid \Omega(U' \cap s, s) < a(s)\}$$

for $U' \subset U$.

PROOF. Suppose that $U' \subset A$ and there exists an element s in $J(U')$ such that $A \cap \mathscr{V}(U', s) = \emptyset$. Then we have

$$\Omega(U' \cap s, s) < a(s). \tag{2.1}$$

Since A is a solution, we have the following inequality for the above s,

$$\begin{aligned} 0 &\leqq \Omega(A \cap s, s) - a(s) \\ &= \Omega((A - U') \cap s, s) + \Omega(U' \cap s, s) - a(s) \quad \text{by } U' \subset A \\ &< \Omega((A - U') \cap s, s) \quad \text{by (2.1)}. \end{aligned}$$

Thus

$$(A - U') \cap s \neq \emptyset.$$

From the assumption, we have $(A - U') \cap \mathscr{V}(U', s) = \emptyset$. If $u \in (A - U') \cap s$, then $u \notin \mathscr{V}(U', s)$ and $u \notin U'$. Then, there is an s' in $\boldsymbol{S}$ such that

$$\Omega((U' \cup \{u\}) \cap s', s') > b(s') \quad \text{for any } u \in (A - U') \cap s$$

by the definition of $\mathscr{V}$. Since $U' \cup \{u\} \subset A$, this contradicts the assumption that A is a solution. (Q.E.D.)

THEOREM 3 (Relationship between a solution of a WSP and the tree $T(P)$). *For a WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$ P, P has a solution, iff there is a node $(i_0, \dots, i_k)$ of $T(P)$ such that*

$$J(\mathscr{A}(i_0, \dots, i_k)) = \emptyset \tag{2.2}$$

PROOF. (Necessity) Let A be a solution of P. Assume that

$$J(\mathscr{A}(i_0, \dots, i_l)) \neq \emptyset \quad \text{for any node } (i_0, \dots, i_l) \text{ of } T(P). \tag{2.3}$$

Since $\mathscr{A}(i_0) = \emptyset$, $\mathscr{A}(i_0) \subset A$. If $A = \emptyset$, then $J(\mathscr{A}(i_0)) = \emptyset$. Then, $|A| \geqq 1$. Moreover assume a node $(i_0, i'_1, \dots, i'_k)$ of $T(P)$ satisfies

$$\mathscr{A}(i_0, i'_1, \dots, i'_k) \subset A \tag{2.4}$$

for any $k \in Z_{+0}$ such that $k < |A|$. Then,

$$A \cap \mathscr{V}(\mathscr{A}(i_0, i'_1, \dots, i'_k), s) \neq \emptyset \quad \text{for any } s \in J(\mathscr{A}(i_0, i'_1, \dots, i'_k))$$

by Lemma 5. There are u' and s' such that

$$u' \in A \cap \mathscr{V}(\mathscr{A}(i_0, i'_1, \dots, i'_k), s').$$

Then, $u' \in \mathscr{V}(\mathscr{A}(i_0, i'_1, \dots, i'_k), s')$. Thus, we have

$$\mathscr{A}(i_0, i'_1, \dots, i'_k) \cap \{u'\} = \emptyset$$

by the definitin of $\mathscr{V}$. Let u' be the first element of the set with some order of all different elements of $\mathscr{V}(\mathscr{A}(i_0, i'_1, \dots, i'_k), s')$ and we have

$$\mathscr{A}(i_0, i'_1, \dots, i'_k, 1) = \mathscr{A}(i_0, i'_1, \dots, i'_k) \cup \{u'\}$$
$$\subset A \qquad \text{by (2 .4) and } u' \in A.$$

Thus,

$$\mathscr{A}(0, 1, \dots, 1) = A.$$

Then

$$J(A) = J(\mathscr{A}(0, 1, \dots, 1)) \neq \emptyset \qquad \text{by (2.3)}$$

, that is, $\Omega(A \cap s, s) < a(s)$. This contradicts that A is a solution of P.

(Sufficiency) Let $(i_0, \dots, i_k)$ be a node of $T(P)$ such that $J(i_0, \dots, i_k)) = \emptyset$, and we have

$$\Omega(\mathscr{A}(i_0, \dots, i_k) \cap s, s) \leqq b(s) \qquad \text{for any } s \text{ in } \boldsymbol{S},$$

and

$$\Omega(\mathscr{A}(i_0, \dots, i_k) \cap s, s) \geqq a(s) \qquad \text{for any } s \text{ in } \boldsymbol{S}$$

by the definition of J. Hence $\mathscr{A}(i_0, \dots, i_k)$ is a solution of P. (Q.E.D.)

From Definitins 5,6 and Theorem 3 for a WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$ P, we can construct the labeled tree $T(P)$. If P has a solution, then there is a node of $T(P)$ in which one of the labels corresponds to a solution of P. Hence we can find a solution to search $T(P)$. In order to search $T(P)$ by using a computer, we should represent Theorem 3 as an algorithm. In the following section, we will propose such an algorithm.

3. An algorithm for solving WSP

An algorithm for constructing a solution of a given WSP P is to search each node of the tree $T(P)$ exhaustively. But in the tree by Definitin 6, there are possibly two or more nodes with the same label which causes redundant searches. Then we will give a theorem to avoid them.

LEMMA 6. *Let P be a WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$ and $T(P)$ be the tree derived from $\mathscr{A}$ on P. If a node $(i_0, \dots, i_k)$ of $T(P)$ satisfies*

$$L((i_0, \dots, i_k), T(P)) \subset D(T(P)),$$

then a descendant $(i_0, \dots, i_k, j_1, \dots, j_m)$ $(m \in \{0, 1, \dots, |U| - k\})$ of $(i_0, \dots, i_k)$ satisfies

$$J(\mathscr{A}(i_0, \dots, i_k, j_1, \dots, j_m)) \neq \emptyset,$$

where

$$D(T(P)) = \{(i_0, \dots, i_k) \in \mathbf{N} \mid (i_0, \dots, i_k) \text{ is a node of } T(P),$$
$$B = \mathscr{A}(i_0, \dots, i_k),$$
$$J(B) \neq \emptyset \text{ and}$$
$$(\forall s \in J(B))(\mathscr{V}(B, s) = \emptyset)\},$$

$$L((i_0, \dots, i_k), T(P))$$
$$= \{(i_0, \dots, i_k, j_1, \dots, j_q) \in \mathbf{N} \mid (i_0, \dots, i_k, j_1, \dots, j_q) \text{ is a node of } T(P),$$
$$q \in \{0, 1, \dots, |U| - k\} \text{ and}$$
$$\mathscr{S}(\mathscr{A}(i_0, \dots, i_k, j_1, \dots, j_q)) = \emptyset\},$$

$$\mathbf{N} = \{0\} \cup (\textstyle\bigcup_{k=1}^{|U|} (\{0\} \times Z_+^k)).$$

PROOF. If a node $(i_0, \dots, i_k)$ of $T(P)$ satisfies

$$L((i_0, \dots, i_k), T(P)) \subset D(T(P))$$

and its descendant $(i_0, \dots, i_k, j_1, \dots, j_m)$ $(m \in \{0, 1, \dots, |U| - k\})$ satisfies

$$J(\mathscr{A}(i_0, \dots, i_k, j_1, \dots, j_m)) = \emptyset,$$

then we have

$$(i_0, \dots, i_k, j_1, \dots, j_m) \in L((i_0, \dots, i_k), T(P))$$

by the definition of L. This contradicts that $L((i_0, \dots, i_k), T(P)) \subset D(T(P))$.
(Q.E.D.)

THEOREM 4. *Let P be a WSP $\langle U, \mathbf{S}, \omega, a, b \rangle$, $T(P)$ be the tree derived from $\mathscr{A}$ on P, and $(i_0, \dots, i_k, j)$ be connected with a node $(i_0, \dots, i_k)$ of $T(P)$ for each j in $\{1, 2, \dots, l\}$. If we have*

$$\mathscr{A}(i_0, \dots, i_k, j') = \mathscr{A}(i_0, \dots, i_k) \cup \{u'\},$$

$$L((i_0, \dots, i_k, j'), T(P)) \subset D(T(P)) \quad \text{for some node } (i_0, \dots, i_k, j'),$$

and

$$\mathscr{A}(i_0, \dots, i_k, j, j_1, \dots, j_m) \ni u'$$

for some element $(i_0, \dots, i_k, j, j_1, \dots, j_m)$ of $L((i_0, \dots, i_k, j), T(P))$ such that $j \neq j'$, then

$$(i_0, \dots, i_k, j, j_1, \dots, j_m) \in D(T(P)).$$

PROOF. Assume that

$$(i_0, \dots, i_k, j, j_1, \dots, j_m) \notin D(T(P)),$$

on these hypothesis. Let U' be $\mathscr{A}(i_0, \dots, i_k, j, j_1, \dots, j_m)$, and we have

$$J(U') = \emptyset. \tag{3.1}$$

Then U' is a solution of P given by Theorem 3. Since $U' \ni u'$, we have

$$\mathscr{A}(i_0, \dots, i_k, j') = \mathscr{A}(i_0, \dots, i_k) \cup \{u'\} \subset U'. \tag{3.2}$$

On the other hand, we have

$$J(\mathscr{A}(i_0, \dots, i_k, j', j'_1, \dots, j'_n)) \neq \emptyset \quad \text{by Lemma 6.}$$

From (3.2) and Lemma 5 there exists a descendant $(i_0, \dots, i_k, j', j'_1, \dots, j'_n)$ of $(i_0, \dots, i_k, j')$ which satisfies

$$\mathscr{A}(i_0, \dots, i_k, j', j'_1, \dots, j'_n) = U'.$$

Then $(i_0, \dots, i_k, j', j'_1, \dots, j'_n)$ is an element of $L((i_0, \dots, i_k, j'), T(P))$ such that $J(\mathscr{A}(i_0, \dots, i_k, j', j'_1, \dots, j'_n)) = \emptyset$ by (3.1). Therefore $(i_0, \dots, i_k, j', j'_1, \dots, j'_n) \notin D(T(P))$. This contradicts that $L((i_0, \dots, i_k, j'), T(P)) \subset D(T(P))$.

(Q.E.D.)

This theorem enable us to avoid redundant searches of the tree $T(P)$.

Now we show an algorithm for construting a solution of a given WSP based on Definitions 5,6 and Theorems 3,4. When a WSP $\langle U, S, \omega, a, b\rangle$ P is given, the following algorithm can search each node with different label.

ALGORITHM 1.

1) Let $n := |U|$, $S := \{s_1, \dots, s_m\}$, $A := \emptyset$, $J_0 := 1$, $i := 1$, $t_j := 0$ for each j in $\{1, \dots, m\}$ and $I_k := \emptyset$, $H_k := \emptyset$ for each k in $\{1, \dots, n\}$.
2) Let $j := J_{i-1}$.
   - 2.1) If $t_j < a(s_j)$, then let $J_i := j$ and go to 3), else go to 2.2).
   - 2.2) Let $j := j + 1$. If $j \leqq m$, then go to 2.1), else **exit**: "A is a solution".
3) If there exists an element u in $s_{J_i} - H_i$, then let $H_i := H_i \cup \{u\}$ and go to 3.1), else go to 3.2).

3.1) If $u \in A \cup I_1 \cup \cdots \cup I_i$, then go to 3), else go to 3.1.1).
   - 3.1.1) If $t_k + \omega(u, s_k) \leqq b(s_k)$ for any s_k including u, then go to 3.1.2), else go to 3).
   - 3.1.2) Let $u_i := u$,
     $A := A \cup \{u_i\}$,
     $t_k := t_k + \omega(u_i, s_k)$ for $\forall s_k \ni u_i$,

$i := i + 1$
go to 2).

3.2) If $i = 1$, then **exit**: "there is no solution".
If $i \geqq 2$, then let $I_i := \emptyset$,
$H_i := \emptyset$,
$I_{i-1} := I_{i-1} \cup \{u_{i-1}\}$,
$A := A - \{u_{i-1}\}$,
$t_k := t_k - \omega(u_{i-1}, s_k)$ for $\forall s_k \ni u_{i-1}$,
$i := i - 1$,
go to 3).

In this algorithm, set A indicates a label of a node of the tree. If a node $(i_0, \ldots, i_k)$ is called a level k node, a level k node has a label $A = \{u_1, \ldots, u_k\}$. Term t_j means a value of $\Omega(A \cap s_j, s_j)$. In step 2), element s with the smallest number of suffix are selected, but any other element of $\boldsymbol{S}$ is also available. Subset I_i contains of all processed elements of U which cannot be included in A.

4. An application to the four color problems

In this section, we show an application to graph coloring problems (say 4CP) defined as follows:

PROBLEM (4CP). For a finite graph $G = (V, E)$, is there a function C from V to $\{1, 2, 3, 4\}$ so that $C(v) \neq C(v')$ if (v, v') belongs to E for any pair (v, v') in V?

DEFINITION 7 (A WSP derived from a 4CP). For a given 4CP $G = (V, E)$, the WSP $\langle U, \boldsymbol{S}, \omega, a, b \rangle$ derived from a G is defined as follows:

$$U = \{(v, c) | v \in V, \ c \in \{1, 2, 3, 4\}\},$$

$$\boldsymbol{S} = \boldsymbol{S}_1 \cup \boldsymbol{S}_2,$$

$$\omega(u, s) = 1 \quad \text{if } u \in s \text{ and } s \in \boldsymbol{S},$$

$$a(s_1(v)) = 1 \quad \text{for any } v \in V,$$

$$b(s_1(v)) = 1 \quad \text{for any } v \in V,$$

$$a(s_2(e, c)) = 0 \quad \text{for any } e \in E \text{ and } c \in \{1, 2, 3, 4\},$$

$$b(s_2(e, c)) = 1 \quad \text{for any } e \in E \text{ and } c \in \{1, 2, 3, 4\},$$

where

$$\boldsymbol{S}_1 = \{s_1(v) | v \in V\}, \tag{4.1}$$

$$S_2 = \{s_2(e, c) | e \in E,\ c \in \{1, 2, 3, 4\}\}, \tag{4.2}$$

$$s_1(v) = \{(v, c) \in U | c \in \{1, 2, 3, 4\}\}, \tag{4.3}$$

$$s_2(e, c) = \{(v_p, c), (v_q, c)\} \text{ for } e = \{v_p, v_q\}. \tag{4.4}$$

THEOREM 5. *If there is a solution of the WSP derived from a 4CP, then a solution of the 4CP can be constructed.*

PROOF. Let P be the WSP $\langle U, S, \omega, a, b\rangle$ derived from a 4CP and A be a solution of P. From (4.1), we have

$$\Omega(A \cap s_1(v), s_1(v)) = |A \cap s_1(v)| = 1 \quad \text{for any } v \in V.$$

For each pair (v, c) in $s_1 \cap A$, we define $C(v) = c$. We have

$$\begin{aligned} 0 &\leqq \Omega(A \cap s_2(e, c), s_2(e, c)) \\ &= |A \cap s_2(e, c)| \\ &\leqq 1 \end{aligned}$$

for any $e \in E$ and any $c \in \{1, 2, 3, 4\}$. This means $C(v) \neq C(v')$ if (v, v') belongs to E for any pair (v, v') in V. (Q.E.D.)

We show an algorithm for constructing a WSP from a 4CP.

ALGORITHM 2.

1) Let $n = |V|$, $m = |E|$, $U := \emptyset$.
2) For i from 1 to n;
   2.1) Let $s_1(v_i) := \emptyset$, $a(s_1(v_i)) := 1$, $b(s_1(v_i)) := 1$,
   2.2) For c from 1 to 4;
   Let $U := U \cup \{(v_i, c)\}$, $s_1(v_i) := s_1(v_i) \cup \{(v_i, c)\}$,
3) For i from 1 to m;
   3.1) For c from 1 to 4;
   Let $s_2(e_i, c) := \{(v_{i_1}, c), (v_{i_2}, c)\}$,
   $a(s_2(e_i, c)) := 0$, $b(s_2(e_i, c)) := 1$,
   where $e_i = (v_{i_1}, v_{i_2})$.

Figure 7 illustrates a graph on a 4CP and Figure 8 provides the WSP derived from the graph in Figure 7. In Figure 8, column A indicates a solution of the WSP. This means that $C(v_1) = 1$, $C(v_2) = 2$, $C(v_3) = 3$, $C(v_4) = 4$ and $C(v_5) = 1$. It is easy to confirm that this function gives a coloring of the 4CP.

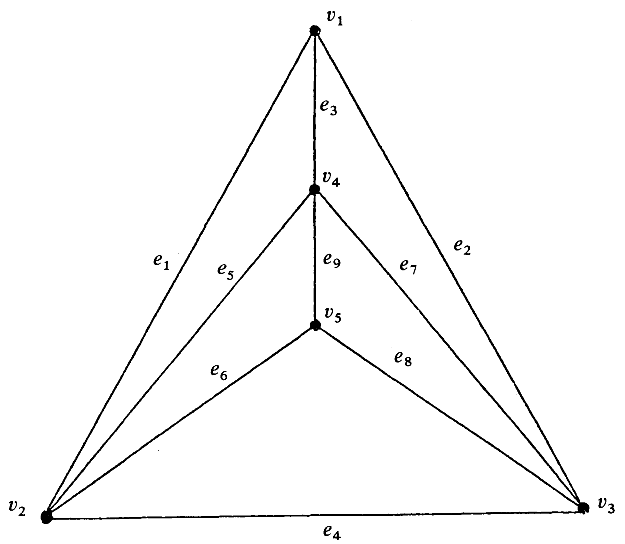


Figure 7 A graph on a 4CP

Acknowledgements

I am very grateful to Professor H. Ikeda of Hiroshima University for guidance of this work with important advice and fruitful discussions. He also helped me sincerely any time when I had some trouble. I am much indebted to Professor M. Sugawara of Hiroshima University for the encouragement and important comments. Acknowledgement also is due to the members of Information Processing Center, Hiroshima University for providing an environment in which I carried out this work successfully.

$\tilde{\omega}$ \ S	S_1					S_2											
						$e_1=(v_1,v_2)$				$e_2=(v_1,v_3)$				$e_3=(v_1,v_4)$			
U	v_1	v_2	v_3	v_4	v_5	c:1	2	3	4	c:1	2	3	4	c:1	2	3	4
$(v_1,1)$	1					1				1				1			
$(v_1,2)$	1						1				1				1		
$(v_1,3)$	1							1				1				1	
$(v_1,4)$	1								1				1				1
$(v_2,1)$		1				1											
$(v_2,2)$		1					1										
$(v_2,3)$		1						1									
$(v_2,4)$		1							1								
$(v_3,1)$			1							1							
$(v_3,2)$			1								1						
$(v_3,3)$			1									1					
$(v_3,4)$			1										1				
$(v_4,1)$				1										1			
$(v_4,2)$				1											1		
$(v_4,3)$				1												1	
$(v_4,4)$				1													1
$(v_5,1)$					1												
$(v_5,2)$					1												
$(v_5,3)$					1												
$(v_5,4)$					1												
a	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
b	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$\tilde{\omega}$ \ S	S_2											
	$e_4=(v_2,v_3)$				$e_5=(v_2,v_4)$				$e_6=(v_2,v_5)$			
U	c:1	2	3	4	c:1	2	3	4	c:1	2	3	4
$(v_1,1)$												
$(v_1,2)$												
$(v_1,3)$												
$(v_1,4)$												
$(v_2,1)$	1				1				1			
$(v_2,2)$		1				1				1		
$(v_2,3)$			1				1				1	
$(v_2,4)$				1				1				1
$(v_3,1)$	1											
$(v_3,2)$		1										
$(v_3,3)$			1									
$(v_3,4)$				1								
$(v_4,1)$					1							
$(v_4,2)$						1						
$(v_4,3)$							1					
$(v_4,4)$								1				
$(v_5,1)$									1			
$(v_5,2)$										1		
$(v_5,3)$											1	
$(v_5,4)$												1
a	0	0	0	0	0	0	0	0	0	0	0	0
b	1	1	1	1	1	1	1	1	1	1	1	1

$\tilde{\omega}$ \ S	S_2												
	$e_7=(v_3,v_4)$				$e_8=(v_3,v_5)$				$e_9=(v_4,v_5)$				A
U	c:1	2	3	4	c:1	2	3	4	c:1	2	3	4	
$(v_1,1)$													1
$(v_1,2)$													
$(v_1,3)$													
$(v_1,4)$													
$(v_2,1)$													
$(v_2,2)$													1
$(v_2,3)$													
$(v_2,4)$													
$(v_3,1)$	1				1								
$(v_3,2)$		1				1							
$(v_3,3)$			1				1						1
$(v_3,4)$				1				1					
$(v_4,1)$	1								1				
$(v_4,2)$		1								1			
$(v_4,3)$			1								1		
$(v_4,4)$				1								1	1
$(v_5,1)$					1				1				1
$(v_5,2)$						1				1			
$(v_5,3)$							1				1		
$(v_5,4)$								1				1	
a	0	0	0	0	0	0	0	0	0	0	0	0	
b	1	1	1	1	1	1	1	1	1	1	1	1	

Figure 8 A WSP derived from a 4CP given in Figure 7

Appendix I.

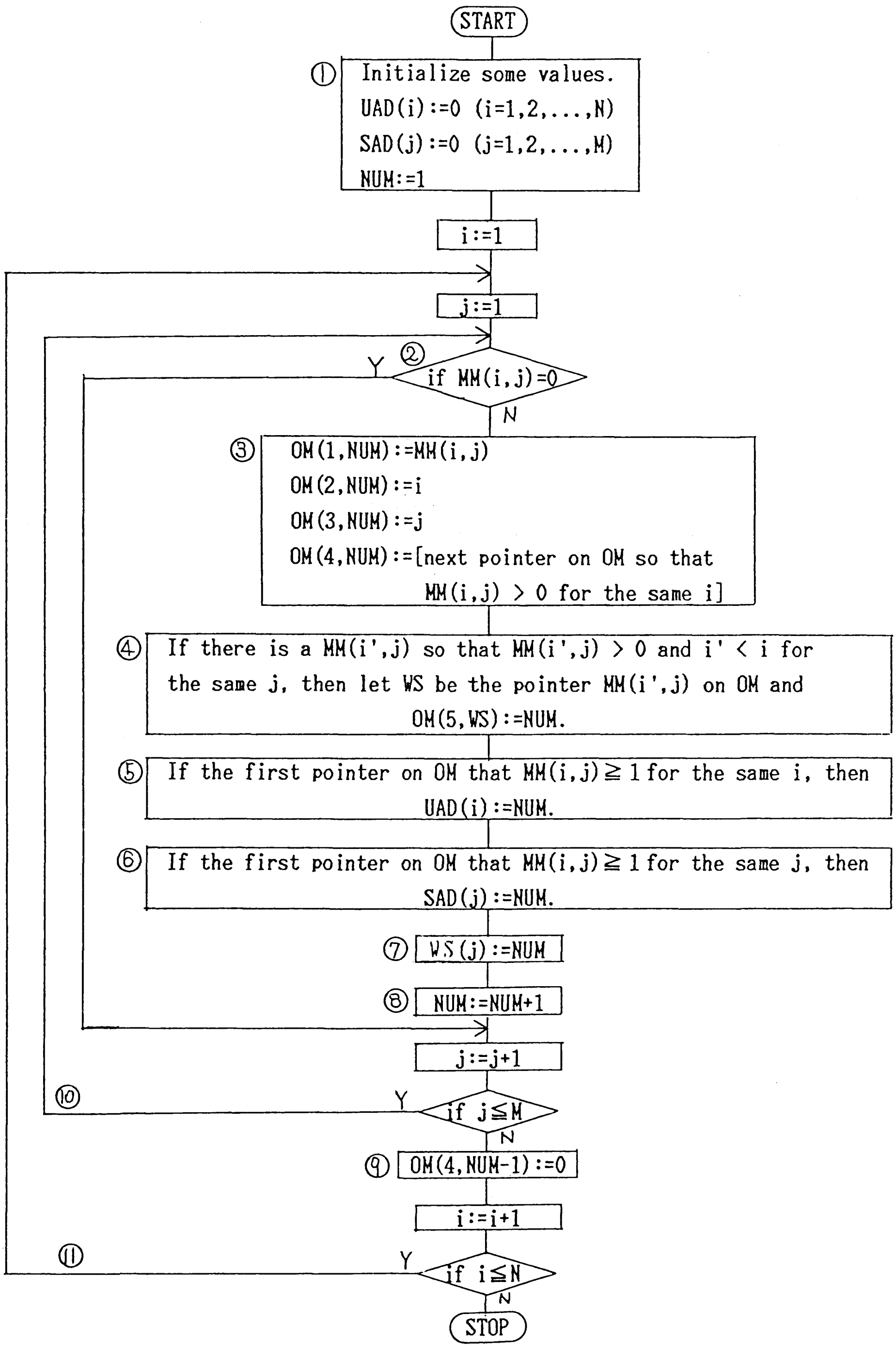
A FORTRAN program given in Appendix I is provided for reduction of an array including 0 values. If the function $\tilde{\omega}$ has many 0 values, then the program is efficient to reduce 0 values in order to require less computer memory than original array. After the specification of parameters, this program are given by both a flow-chart and FORTRAN language.

Explanation of parameters.

MM(i, j): two-dimensional array which corresponds to the incidence matrix of $\tilde{\omega}$ on $U \times S$.
M: the size of each column of array MM (the cardinality of S).
N: the size of each row of array MM (the cardinality of U).
UAD(i): a pointer to table OM to search MM (i, j) in ascending order of j.
SAD(j): a pointer to table OM to search MM (i, j) in ascending order of i.
OM(k, p): a reduced array. If MM $(i, j) = 0$ for some i and j, then this table does not include a column such that OM $(2, p) = i$ and OM$(3, p) = j$ for some p.

	$k = 1$	2	3	4	5
$p =$	MM(i, j)	i	j	next pointer on OM so that MM$(i, j) > 0$ for the same i	next pointer on OM so that MM$(i, j) > 0$ for the same j
1					
2					
⋮					

WS(j): work array to find the value of OM $(5, p)$.
NUN: pointer of OM(k, p)


START
① Initialize some values.
UAD(i):=0 (i=1,2,...,N)
SAD(j):=0 (j=1,2,...,M)
NUM:=1
i:=1
j:=1
② if MM(i,j)=0
Y
N
③ OM(1,NUM):=MM(i,j)
OM(2,NUM):=i
OM(3,NUM):=j
OM(4,NUM):=[next pointer on OM so that
MM(i,j) > 0 for the same i]
④ If there is a MM(i',j) so that MM(i',j) > 0 and i' < i for
the same j, then let WS be the pointer MM(i',j) on OM and
OM(5,WS):=NUM.
⑤ If the first pointer on OM that MM(i,j)≧1 for the same i, then
UAD(i):=NUM.
⑥ If the first pointer on OM that MM(i,j)≧1 for the same j, then
SAD(j):=NUM.
⑦ WS(j):=NUM
⑧ NUM:=NUM+1
j:=j+1
⑩
Y
if j≦M
N
⑨ OM(4,NUM-1):=0
i:=i+1
⑪
Y
if i≦N
N
STOP

```
SUBROUTINE SPCONV(MM)
COMMON OM,SI,TT,A,B,UI,AN,N,M,SAD,UAD
INTEGER OM(5,0:20000),SI(0:201),TT(200)
INTEGER A(200),B(200),UI(101),AN(2,100)
INTEGER N,M,SAD(200),UAD(100)
INTEGER MM(100,200),WS(200)
"
      DO 1 I=1,N                                        ①
         UAD(I)=0
    1 CONTINUE
      DO 2 I=1,M
         SAD(I)=0
         WS(I)=0
    2 CONTINUE
      NUM=1
      DO 10 I=1,N                                          ⑪
         DO 11 J=1,M                                    ⑩
            IF (MM(I,J).EQ.0) GOTO 11 ---------------- ②
               OM(1,NUM)=MM(I,J)                        ③
               OM(2,NUM)=I
               OM(3,NUM)=J
               OM(4,NUM)=NUM+1
               IF (WS(J).GT.0)   OM(5,WS(J))=NUM ----- ④
               IF (UAD(I).EQ.0) UAD(I)=NUM ----------- ⑤
               IF (SAD(J).EQ.0) SAD(J)=NUM ----------- ⑥
               WS(J)=NUM ----------------------------- ⑦
               NUM=NUM+1 ----------------------------- ⑧
   11    CONTINUE
      OM(4,NUM-1)=0 ---------------------------------- ⑨
   10 CONTINUE
      RETURN
      END
```

Appendix II.

A tree searching program given in Appendix II is provided for solving WSP based on Algorithm 1. Before showing a flow-chart and its FORTRAN program, the explanation of parameters are given as follows:

Explanation of parameters.

OM (i, j), SAD (j), UAD (i), M and N are the same of previous explanations.

AN (p, q): This array has two parts. One part is a set A corresponding to function $\mathscr{A}(k_0, \ldots, k_i)$, and the other part is each set I_i to avoid redundant searches. If AN (1, q) $\geqq 1$, it means that q is included in A. If AN (1, q) $= -i$, it means that q is included in I_i. AN (2, q) is used for removing of I_i effectively.

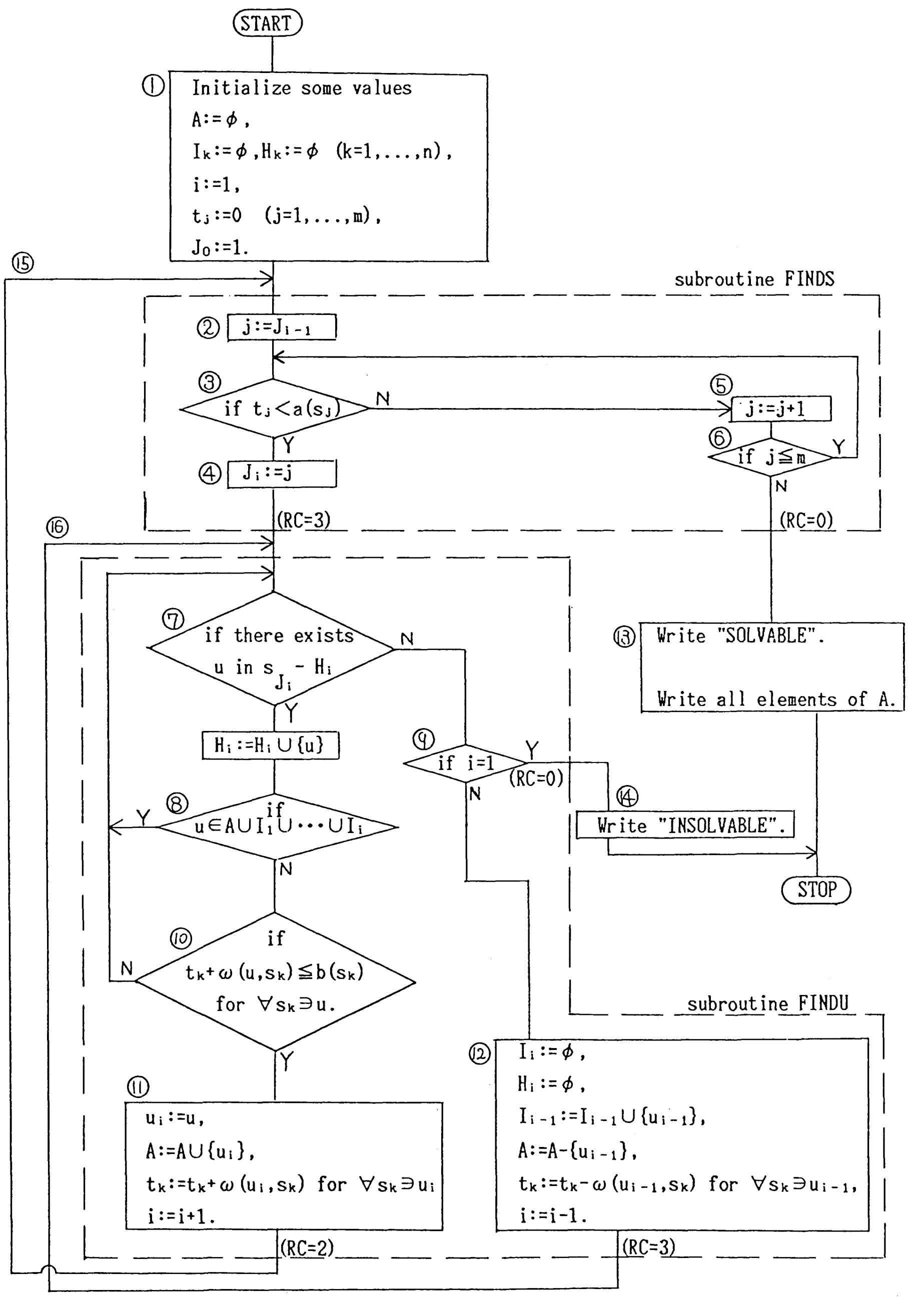
TT (j): corresponds to t_j in Algorithm 1, that is, $\Omega(A \cap s_j, s_j)$.

SI (i): corresponds to J_i in Algorithm 1.

UI (i): corresponds to u_i in Algorithm 1.

A (j): corresponds to the lower bound function a for s_j.

B(j): corresponds to the upper bound function b for s_j.


START
① Initialize some values
A:=φ,
I_k:=φ, H_k:=φ (k=1,...,n),
i:=1,
t_j:=0 (j=1,...,m),
J_0:=1.
⑮
subroutine FINDS
② j:=J_{i-1}
③ if t_j<a(s_j)
N
Y
④ J_i:=j
⑤ j:=j+1
⑥ if j≦m
Y
N
(RC=3)
(RC=0)
⑯
⑦ if there exists u in s_{J_i} - H_i
N
Y
H_i:=H_i ∪ {u}
⑧ if u∈A∪I_1∪···∪I_i
Y
N
⑨ if i=1
Y
(RC=0)
N
⑩ if t_k+ω(u,s_k)≦b(s_k) for ∀s_k∋u.
N
Y
⑪ u_i:=u,
A:=A∪{u_i},
t_k:=t_k+ω(u_i,s_k) for ∀s_k∋u_i
i:=i+1.
(RC=2)
⑫ I_i:=φ,
H_i:=φ,
I_{i-1}:=I_{i-1}∪{u_{i-1}},
A:=A-{u_{i-1}},
t_k:=t_k-ω(u_{i-1},s_k) for ∀s_k∋u_{i-1},
i:=i-1.
(RC=3)
subroutine FINDU
⑬ Write "SOLVABLE".
Write all elements of A.
⑭ Write "INSOLVABLE".
STOP

```
SUBROUTINE SPTREE
COMMON OM,SI,TT,A,B,UI,AN,N,M,SAD,UAD
INTEGER OM(5,0:20000),SI(0:201),TT(200)
INTEGER A(200),B(200),UI(101),AN(2,100)
INTEGER RC,N,M,SAD(200),UAD(100)
REAL*16 COUNT
"Initialize some values
      COUNT=0                  ┐
      SI(0)=1                  │
      I=1                      │
      DO 10 J=1,M              │
         TT(J)=0               │
   10 CONTINUE                 │
      DO 20 J=1,N              ├①
         DO 21 K=1,2           │
            AN(K,J)=0          │
   21    CONTINUE              │
         UI(J)=0               │
   20 CONTINUE                 │
      UI(N+1)=0                ┘
"Start of finding a solution
    2 CONTINUE
      CALL FINDS(I,RC)
      IF (RC.EQ.0) GOTO 888
    3 CONTINUE
      CALL FINDU(I,RC,COUNT)
      IF (RC.EQ.2) GOTO 2 ——⑮
      IF (RC.EQ.3) GOTO 3 ——⑯
      WRITE(*,400) COUNT                                  ┐
  400 FORMAT(1X,'INSOLVABLE, SEARCH NUMBER =',F10.0)      ┘⑭
      GOTO 999
  888 CONTINUE
      WRITE(*,*) 'SOLVABLE'                     ┐
      DO 30 K=1,N                               ├⑬
         IF (AN(1,K).GT.0) WRITE(*,200) K       ┘
   30 CONTINUE
      WRITE(*,300) COUNT
  200 FORMAT(1X,'ANSWER =',I4)
  300 FORMAT(1X,'TOTAL NUMBER OF SEARCH =',F10.0)
  999 CONTINUE
      RETURN
      END
SUBROUTINE FINDS(I,RC)
COMMON OM,SI,TT,A,B,UI,AN,N,M,SAD,UAD
INTEGER OM(5,0:20000),SI(0:201),TT(200)
INTEGER A(200),B(200),UI(101),AN(2,100)
INTEGER S,N,M,RC,SAD(200),UAD(100)
" Search s which satisfies TT(s)<A(s)
      J=SI(I-1) ————————————————————②
      DO 10 S=J,M                              ┐
         IF (TT(S).LT.A(S)) GOTO 50 —③         ├⑤⑥
   10 CONTINUE                                 ┘
      RC=0
      RETURN
```

```
"If the problem has a solution, this point is the END of search.
"
   50 CONTINUE
"Initialize some values for subroutine FINDU
      SI(I)=S          ⎫
      UI(I)=0          ⎪
      OM(5,0)=SAD(S)   ⎬④
      OM(2,0)=0        ⎭
      RC=3
      RETURN
      END


SUBROUTINE FINDU(I,RC,COUNT)
COMMON OM,SI,TT,A,B,UI,AN,N,M,SAD,UAD
INTEGER OM(5,0:20000),SI(0:201),TT(200)
INTEGER A(200),B(200),UI(101),AN(2,100)
INTEGER U,RC,N,M,SAD(200),UAD(100)
REAL*16 COUNT
"Search u which is in OM(U,S(I)) and not in AN
      U=UI(I)
    1 CONTINUE
         U=OM(5,U)
         IF(U.EQ.0) GOTO 2 ----------------⑦
         IF (AN(1,OM(2,U)).EQ.0) GOTO 50 -⑧
         GOTO 1
    2 CONTINUE
      IF (I.LE.1) THEN --------------------⑨
         RC=0
         RETURN
      END IF
"Trace Back when it has no solution.
      IF (UI(I).EQ.0) GOTO 30          ⎫
      K=OM(2,UI(I))                    ⎪
   31 CONTINUE                         ⎪
         J=AN(2,K)                     ⎪
         AN(1,K)=0                     ⎪
         AN(2,K)=0                     ⎪
         IF (J.EQ.0) GOTO 30           ⎪
         K=J                           ⎪
         GOTO 31                       ⎪
   30 CONTINUE                         ⎪
      I=I-1                            ⎪
      U=OM(2,UI(I))                    ⎬⑫
      AN(1,U)=-(I)                     ⎪
      K=UAD(U)                         ⎪
   21 CONTINUE                         ⎪
      IF (K.EQ.0) GOTO 20              ⎪
         J=OM(3,K)                     ⎪
         TT(J)=TT(J)-OM(1,K)           ⎪
         K=OM(4,K)                     ⎪
         GOTO 21                       ⎭
   20 CONTINUE
      RC=3
      RETURN
```

```
"
"Check u if TT(s)+OM(U,s)>B(s) for all s
   50 CONTINUE                                         ⎫
      J=UAD(OM(2,U))                                   ⎪
   51 CONTINUE                                         ⎪
      IF (J.EQ.0) GOTO 60                              ⎬⑩
         K=OM(3,J)                                     ⎪
         IF ((TT(K)+OM(1,J)).GT.B(K)) GOTO 1           ⎪
         J=OM(4,J)                                     ⎪
         GOTO 51                                       ⎭
   60 CONTINUE
"
      J=OM(2,UI(I))                                    ⎫
      UI(I)=U                                          ⎪
      K=OM(2,U)                                        ⎪
      AN(1,K)=I                                        ⎪
      AN(2,K)=J                                        ⎪
      K=UAD(K)                                         ⎪
   71 CONTINUE                                         ⎬⑪
      IF (K.EQ.0) GOTO 70                              ⎪
         J=OM(3,K)                                     ⎪
         TT(J)=TT(J)+OM(1,K)                           ⎪
         K=OM(4,K)                                     ⎪
         GOTO 71                                       ⎪
   70 CONTINUE                                         ⎪
      I=I+1                                            ⎭
      COUNT=COUNT+1
      RC=2
      RETURN
      END
```

References

[1] M. Almond, An algorithm for constructing university time-tables, Comp. J., **8** (1966), 331–340.

[2] K. Appel and W.Haken, Every planar map is four colorable, Illinois J. of math., **21** (3) (1977).

[3] J. S. Appleby, D. V. Blake and E. A. Newman, Techniques for producing school timetables on a computer and their application to other scheduling problems, Comp. J., **3** (5) (1961) 237–245.

[4] E. D. Barraclough, The application of a digital computer to the construction of timetables, Comp. J., **8** (1965) 136–146.

[5] A. J. Cole, The preparation of examination timetables using a small-store computer, Comp. J., **7** (1964) 117-121.

[6] J. Csima and C. C. Gotlieb, Tests on a computer method for constructing school timetables, Comm. ACM, **7** (3) (1964) 160–163.

[7] M. A. H. Dempster, Two algorithms for the timetable problem, Combinatorial Theory and its Applications (Academic Press, London, 1971) 63–85.

[8] D. De Werra, Construction of timetables by flow methods, INFOR. J., **9** (1) (1971) 12–22.
[9] A. K. Duncan, Further results on a computer construction of school timetables, Comm. ACM, **8** (1) (1965) 72.
[10] C. C. Gotlieb, The construction of class-teacher timetables, Proc. IFIP Congr. Munich (North Holland, Amsterdam, 1962) 73–77.
[11] H. Ikeda, S. Itoh and S. Yamamoto, An algorithm for school timetable construction, Proceedings of the 15th Conf. on Information Processing Society of Japan (IPSJ) (1974) 553–554 (Japanese).
[12] H. Ikeda and Y. Yamashita, A simultaneous inequality model of school timetable, 18th IPSJ (1977) 623–624 (Japanese).
[13] K. Kirchgassner, Some algorithms for solving parts of the class scheduling problem (Internal Research Report, M. I. T. 1963).
[14] F. Kitagawa and H. Ikeda, An existential problem of 0–1 matrix and its application to school timetable construction, 29th IPSJ (1984) 1807–1808 (Japanese).
[15] F. Kitagawa and H. Ikeda, A model of timetable which includes many complex conditions, 32nd IPSJ (1986) 2131–2132 (Japanese).
[16] F. Kitagawa and H. Ikeda, An existential problem of a weight-controlled subset and its application to school time-table construction, Discrete math. **72** (1988) 195–211.
[17] J. Lions, Matrix reduction using the Hungarian method for the generation of school timetables, Comm. ACM, **9** (5) (1966) 349–354.
[18] J. Lions, A counter-example for Gotlieb's method for the construction of school timetables, Comm. ACM, 9 (1966) 697–698.
[19] J. Lions, The Ontario school scheduling program, Comp. J., **10** (1967) 14–20.
[20] G. A. Neufeld and J. Tarter, Graph colouring condition for the existence of solutions to the timetable problem, Comm. ACM, **17** (8) (1974) 450–453.
[21] K. Ohmori, Latin square, (Toyamabo 1973) (Japanese).
[22] A. Salazar and R. V. Oakford, A graph formulation of a school scheduling algorithm, Comm. ACM, **17** (12) (1974) 696–698.
[23] D. J. A. Welsh and M. B. Powell, An upper bound to the chromatic number of a graph and its application to timetabling problems, Comp. J., **10** (1) (1967) 85–86.
[24] D. C. Wood, A system for computing university examination timetables, Comp. J., **11** (1968) 41–47.
[25] S. Yamamoto, H. Ikeda, S. Sige-eda, K. Ushio and N. Hamada, On Claw-decomposition of complete graphs and cmplete bi-graphs, Hiroshima Math. J., **5** (1975) 33–42.
[26] A. P. Yule, Extensions to the heuristic algorithm for university timetables, Comp. J., **10** (1967) 360–364.

Information Processing Center,
Hiroshima University