A Book-keeping Method for the Solution of the Consistent Labeling Problem

XiaoYou Zhou and Wayne A. Davis

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1

ABSTRACT

This paper introduces a new method for the solution of the consistent labeling problem. This method aims at increasing the efficiency of tree search by using a look-ahead operator. This new approach is such that unnecessary repetitive computations are avoided at a cost of extra memory for book-keeping.

Key Words: consistent labeling, constraint satisfaction, look-ahead operator, relaxation operator, scene analysis, tree search.

1. Introduction

The general computational problem of assigning labels consistently to objects is called the "consistent labeling problem". The problem is a generalization of specific problems from several specialty areas such as graph and automata homomorphism, graph coloring, Latin square generation, image understanding, and theorem proving [1-3,5,7].

A variety of models have been developed for the solution of the problem since the 1970's [5-7]. In this paper, the (R,T)-consistent model, proposed by Haralick and Shapiro [6], is used throughout. This model is defined as follows:

Definition 1.: A (R,T)-consistent model is a quadruple (U,L,T,R) where

1). $U = \{1,...,M\}$ is a set of units which represent the set of objects to be labeled.

2). L is the set of labels to be assigned to the units.

3). $T \subseteq U^N$ is the set of all N-tuples of units which mutually constrain one another.

4). $R \subseteq (U \times L)^N$ is the set of constraint relations of all N-tuples of pairs $(u_1,l_1,\ldots,u_N,l_N)$ where $(l_1,\ldots,l_N)$ is a legal labeling of units $(u_1,\ldots,u_N)$.

5). A labeling $(l_1,\ldots,l_p)$ is a consistent labeling of units $(u_1,\ldots,u_p)$ with respect to the compatibility model (U,L,T,R) if and only if $(i_1,\ldots,i_N) \subseteq \{1,...,p\}$ and $(u_{i_1},...,u_{i_N}) \subseteq T$ imply the N-tuples of pairs $(u_{i_1},l_{i_1},\ldots,u_{i_N},l_{i_N}) \subseteq R$; that is, the labeling $(l_{i_1},\ldots,l_{i_N})$ is legal labeling of units $(u_{i_1},\ldots,u_{i_N})$.

The consistent labeling problem is to find all the consistent labelings of units $(1,...,M)$ with respect to the model.

A straightforward way for finding consistent labeling is that of a depth first search procedure. The procedure fixes a label l_h to unit h at level h in the tree and checks if $(l_1,\ldots,l_h)$ is a consistent labeling of $(1,...,h)$ with respect to (T,R). If so then proceed to the next level, otherwise backtrack. This method suffers from thrashing: a poor choice of labels for one of the first units causes failure of all paths stemming from that choice. To overcome this problem, those paths containing no consistent labelings must be eliminated. To this end, look-ahead operators are often used for pre-checking, which is defined as follows:

Definition 2: Let $U=\{1,...,M\}$ be a set of units, L be a set of labels, $T \subseteq U^N$, and $R \subseteq (U \times L)^N$. Let $K \le N \le P$ with $K < P$. The look-ahead operator ϕ_{KP} is defined by $\phi_{KP}R = \{(u_1,l_1,\ldots,u_N,l_N) \in R \mid$ for every combination $j_1,\ldots,j_k$ of $1,...,N$ and for every $u'_{k+1},\ldots,u'_P \in U$, There exists $l'_{k+1},\ldots,l'_P \in L$ such that $(l_{j_1},\ldots,l_{j_k},l'_{K+1},\ldots,l'_P)$ is a (T,R)-consistent labeling of $(u_{j_1},\ldots,u_{j_k},u'_{K+1},\ldots,u'_P)\}$.

The following notational conventions are adopted in this paper:

$\phi_{KP}{}^{\infty}R = \bigcap_{m=1}^{\infty} \phi_{KP}{}^{m}R$.

$R|_{i,l} = \{(u_1,l_1,\ldots,u_N,l_N) \mid (u_1,l_1,\ldots,u_N,l_N) \in R$ and $u_q = i$ implies $l_q = l\}$.

The tree search with the ϕ_{KP} incorporated, fixes label l_h to unit h at level h, calculates $R_n = R$ restricted to those N-tuples of pairs where l_i is the label for unit i, $i=1,...,n$, and applies ϕ_{KP} to R_n until a fixed point is reached. If the fixed point is an empty set then the current path is abandoned, and the procedure backs up. If the fixed point is a single valued relation set then a consistent labeling is found. The procedure can record its answer and either quit or continue looking for more consistent labels. If neither of the above two cases is encountered then the procedure must search further down the tree.

Although the look-ahead operator approach does help in eliminating unnecessary backtracking, it does not take advantage of the previous computation for each subsequent step of

This research was supported in part by NSERC Grant A7634.

the tree search. Thus a lot of effort is spent on brutal repetitive computations[1].

To illustrated the problem, let consider the following example:

Let U={1,2,3,4,5}, L={a,b,c,d}, $T=U\times U$,
R={(2,a,3,b), (2,a,4,c), (2,a,5,d), (2,d,3,a), (2,d,3,c), (2,d,4,b), (2,d,5,a), (3,c,4,c), (3,c,5,a), (3,c,5,d), (3,a,4,b), (3,a,5,a), (3,b,4,c), (3,b,5,d), (4,b,5,a), (4,c,5,d), (2,a,1,a), (2,a,1,b), (3,a,1,a), (3,a,1,b), (3,b,1,a), (3,b,1,b), (3,c,1,a), (3,c,1,b), (2,d,1,a), (2,d,1,b), (4,c,1,a), (4,c,1,b), (4,b,1,a), (4,b,1,b), (5,a,1,a), (5,d,1,a), (5,a,1,b), (5,d,1,b)},
and let the look-ahead operator is ϕ_{13}, R_i denote the current relation set at level i, e.g. R_0=R.

Suppose the tree search starts at unit 1. First fix 1 to a, yielding R_1, see Figure 1.

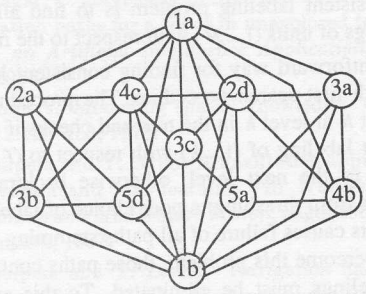

Figure 1. R_0 set.

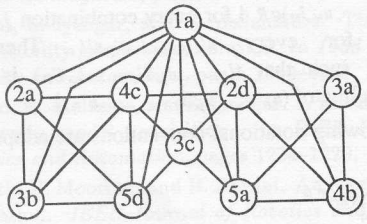

Figure 2. R_1 set.

Since R_1 is already a fixed point of ϕ_{13}, we further fix 2 to a, yielding Figure 3.

[1] readers are referred to [5] for the tree search algorithm.

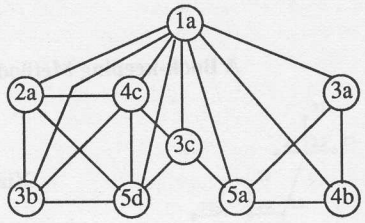

Figure 3. R_2 set.

After applying ϕ_{13} to the fixed point, the first consistent labeling is obtained (1,a,2,a,3,b,4,c,5,d), see Figure 4.

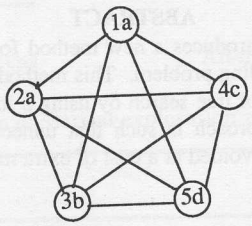

Figure 4. $\phi_{13}{}^{\infty}R_2$ set.

The process then backtracks to upper level and fixs 2 to d, yielding Figure 5.

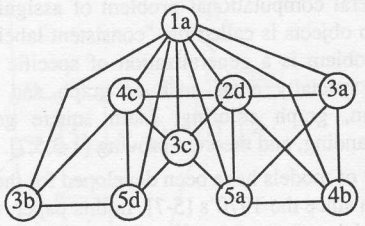

Figure 5. R_2 set.

After applying ϕ_{13} to the fixed point, the second consistent labeling (1,a,2,d,3,a,4,b,5,a) is found, as shown in Figure 6.

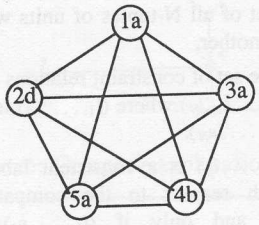

Figure 6. $\phi_{13}{}^{\infty}R_2$ set.

Then the process will backtrack to level 0 and fix 1 to b. The rest of the tree search will be carried out exactly the same way as previously due to the symmetry property of R. From the diagram of R, it is easy to see that the following search will involve with lot of repetitive computations.

The problem with the above approach is that:

1). After the first two applications of ϕ_{13}, it is clearly seen that (2,a,3,b), (2,a,4,c), (2,a,5,d), (3,b,4,c), (3,b,5,d), (4,c,5,d) constitute a clique of size 4. Thus, any subsequent attempt for verifying the same fact should be discouraged.

2). Some N-tuple of pairs, after a period of tree search, have been proved cannot contribute a consistent labeling under any circumstance. Thus, any subsequent effort at trying to generate consistent labeling from those relations should also be avoided.

However, the current implementation does not provide a mechanism for dealing with the situations illustrated above. Thus, repetitive computations is unavoidable. The repetitive computation involved in this example are:

1). The subsequent attempt to test the consistency of (2,a,3,b), (2,a,4,c), (2,a,5,d), (3,b,4,c), (3,b,5,d), (4,c,5,d), and (2,d,3,a), (2,d,4,b), (2,d,5,a), (3,a,4,b), (3,a,5,a), (4,b,5,a).

2). The subsequent attempt to show (4,c,2,d), (2,d,3,c), (2,d,5,a), (3,c,4,c), (3,c,5,a), (3,d,5,d) contribute a consistent labeling.

Since, in general, the test of one relation for its consistency is a costly combinatorial computation, repetitive computation is an obstacle for achieving the most efficient algorithm.

2. A Book-keeping Approach

One way to avoid the repetitive computation is by means of book-keeping. Regard the process of tree search as equivalent to a sequence of prove or disprove under certain circumstances. What is needed is to define and record the circumstances effectively, so that no more calculation is necessary if the search leads to the same case.

To this end, a theorem, called the independent computation theorem, will be given:

Lemma 2.1: Let (U,L,T,R) be a consistent model $R'\subseteq R$, $\bar{R}\subseteq R$, and ϕ_{KP} be a look-ahead operator, then the following claims are true:

a). $\phi_{KP}{}^{\infty}R'\subseteq R'$,

b). $\phi_{KP}(R'\cap\bar{R})=(\phi_{KP}R')\cap(\phi_{KP}\bar{R})$

c). If $\phi_{KP}{}^{\infty}R'=R'$ and $R'\subseteq\bar{R}$ then $R'\subseteq\phi_{KP}{}^{\infty}\bar{R}$.

Proof: Directly derivable from the definition of ϕ_{KP}, see also [5].

Q.E.D.

Lemma 2.2: Let $I=\bigcap_{j=1}^{m'}\phi_{KP}{}^{\infty}(R\mid_{u_j,l_j})$ then $\phi_{KP}(I)=I$, for any $m'\geq 1$ and ϕ_{KP}.

Proof: Directly derivable from **Lemma 2.1.b** and the definition of ϕ_{KP}.

Q.E.D.

Lemma 2.3: Let (U,L,T,R) be consistent model, $I\subseteq R$ and $I'\subseteq R$, then $I\subseteq I'$ and $I\mid_{u,l}=I$ implies $I\subseteq I'\mid_{u,l}$.

Proof: Direct result from the definition.

Q.E.D.

Theorem 2.1(Independent Computation Theorem): Let (U,L,T,R) be a consistent model, $S_q=<(u_1,l_1),\ldots,(u_t,l_t)>$ be a sequence of searching steps which leads R to R_t: $R_t=\phi_{KP}{}^{\infty}[\cdots\phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R\mid_{u_1,l_1})]\mid_{u_2,l_2}\ldots\mid_{u_t,l_t}]$. Then

$$R_t=\bigcap_{j=1}^{t}\phi_{KP}{}^{\infty}(R\mid_{u_j,l_j}).$$

Proof: First, we show $R_t\subseteq\bigcap_{i=1}^{t}(\phi_{KP}{}^{\infty}R\mid_{u,l})$ by induction:

If $t=1$, then $R_1=\phi_{KP}{}^{\infty}(R\mid_{u_1,l_1})=\bigcap_{t=1}^{1}\phi_{KP}{}^{\infty}(R\mid_{u_1,l_1})$.

Suppose the statement is true for $t=\beta$, we shall show it is also true for $t=\beta+1$.

First, for any $(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon R_{\beta+1}$,

$$(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon R_{\beta+1}\rightarrow(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon\phi_{KP}{}^{\infty}(R_R\mid_{u_{\beta+1},l_{\beta+1}})$$
$$\rightarrow(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon R_\beta\mid_{u_{\beta+1},l_{\beta+1}}$$
$$\rightarrow(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon R_\beta$$

Induction hypothesis,

$$(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon\bigcap_{i=1}^{\beta}\phi_{KP}{}^{\infty}(R\mid_{u,l}) \qquad (A)$$

On the other hand,

$R_\beta\subseteq R\rightarrow R_{\beta+1}\subseteq R\mid_{u_{\beta+1},l_{\beta+1}}$.

and,

$(\phi_{KP}{}^{\infty}(R_{\beta+1})=R_{\beta+1})\cap(R_{\beta+1}\subseteq R\mid_{u_{\beta+1},l_{\beta+1}})\rightarrow R_{\beta+1}\subseteq\phi_{KP}{}^{\infty}(R\mid_{u,l})$ (By **Lemma 2.1.c**).

Hence,

$$(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon\phi_{KP}{}^{\infty}(R\mid_{u,l}) \qquad (B)$$

Combining (A) and (B) yields:

$$(u'_1,l'_1,\ldots,u'_N,l'_N)\varepsilon\bigcap_{i=1}^{\beta+1}\phi_{KP}{}^{\infty}(R\mid_{u,l})$$

Therefore, $R_t\subseteq\bigcap_{i=1}^{t}\phi_{KP}{}^{\infty}(R\mid_{u,l})$ is true in general.

Secondly, show $\bigcap_{i=1}^{t}\phi_{KP}{}^{\infty}(R\mid_{u,l})\subseteq R_t$

Let $I=\bigcap_{i=1}^{t}\phi_{KP}{}^{\infty}(R\mid_{u,l})$, then $I\mid_{u_j,l_j}=I$, $I\subseteq R$, and $\phi_{KP}{}^{\infty}I=I$ are true for I, where $j=1,\ldots,t$.

Thus,

$$(I\subseteq R)\cap(I\mid_{u_1,l_1}=I)\cap(\phi_{KP}{}^{\infty}I=I)\rightarrow(I\subseteq R\mid_{u_1,l_1})\cap(\phi_{KP}{}^{\infty}I=I)$$
$$\rightarrow I\subseteq R_1.$$

Suppose $I \subseteq R_\beta$ is true for $1 \le \beta < t$, now show $I \subseteq R_{\beta+1}$.

$$(I \subseteq R_\beta) \cap (I \mid_{u_{\beta+1}, l_{\beta+1}} = I) \rightarrow (I \subseteq R_\beta \mid_{u_{\beta+1}, l_{\beta+1}}) \cap (I \mid_{u_{\beta+1}, l_{\beta+1}} = I)$$
$$\rightarrow I \subseteq \phi_{KP}{}^{\infty}(R_\beta \mid_{u_{\beta+1}, l_{\beta+1}})$$
$$\rightarrow I \subseteq R_{\beta+1}$$

Combining the first part and the second part yields:

$$R_t = \bigcap_{j=1}^{t} \phi_{KP}{}^{\infty}(R \mid_{u_j, l_j})$$

Q.E.D.

The Independent Computation Theorem implies that the fixed point behavior of the tree search is independent to the order of the search sequence. Thus the following corollary is obviously true:

Corollary 2.1: Let $S_q = <(u_1, l_1), \ldots, (u_t, l_t)>$ be a sequence of search steps which leads R to R_t, e.g. $R_t = \phi_{KP}{}^{\infty}[\ldots \phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R \mid_{u_1, l_1}) \mid_{u_2, l_2}] \ldots] \mid_{u_t, l_t}]$, and $<(u'_1, l'_1), \ldots, (u'_t, l'_t)>$ is a permutation of S_q. Then

$$R_t = \phi_{KP}{}^{\infty}[\ldots \phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R \mid_{u'_1, l'_1}) \mid_{u'_2, l'_2}] \ldots] \mid_{u'_t, l'_t}].$$

Definition 3: The state space of (U, L, T, R) is defined as:

$Q = \{(\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M) \mid$ where $\alpha_i \subseteq L$ and $\beta_i \subseteq L - \alpha_i, i = 1, \ldots, M\}$

Given an element $(u_1, l_1, \ldots, u_N, l_N) \varepsilon R$, $(\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)$ is said to be a state of $(u_1, l_1, \ldots, u_N, l_N)$ with respect to ϕ_{KP} if the following conditions are satisfied:

1). For any e, $1 \le e \le M$, such that $i_1, \ldots, i_e$ is a combination permutation of $1, \ldots, M$. Then

$$(u_1, l_1, \ldots, u_N, l_N) \varepsilon \bigcap_{j=1}^{e} \phi_{KP}{}^{\infty}(R \mid_{i_j, l_k}) \text{ for any } l_{i_j} \varepsilon \alpha_{i_j},\ j = 1, \ldots, e.$$

2). For any $l'_i \varepsilon \beta_i$, $i = 1, \ldots, M$:

$$(u_1, l_1, \ldots, u_N, l_N) \notin \phi_{KP}{}^{\infty}(R \mid_{i, l'_i})$$

If $(\alpha_1/\beta_1, \alpha_2/\beta_2, \ldots, \alpha_M/\beta_M)$ is a state of $(u_1, l_1, \ldots, u_N, l_N)$, then denote as:

$$(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M).$$

Furthermore, $<(u_1, l_1, \ldots, u_N, l_N), (\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)>$ is called a relation-state pair with respect to ϕ_{KP}.

Lemma 2.4: For any $(\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)$, $(\alpha'_1/\beta'_1, \ldots, \alpha'_M/\beta'_M)$ such that

$$(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M) \text{ and}$$
$$(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha'_1/\beta'_1, \ldots, \alpha'_M/\beta'_M)$$

Then

$$(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow ((\alpha_1 \cup \alpha'_1)/(\beta_1 \cup \beta'_1), \ldots, (\alpha_M \cup \alpha'_M)/(\beta_M \cup \beta'_M))$$

Proof: Directly derivable from the definition.

Q.E.D.

Lemma 2.5: If $(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)$ and $\alpha_i \neq \phi$ for any $l'_i \varepsilon \alpha_i$, $i = 1, \ldots, M$. Then $(l'_1, \ldots, l'_N)$ is a consistent labeling.

Proof: Let $l'_i \varepsilon \alpha_i$, for $1 \le i \le M$, Then

$$(u_1, l_1, \ldots, u_N, l_N) \varepsilon \bigcap_{i=1}^{M} \phi_{KP}{}^{\infty}(R \mid_{i, l'_i})$$
$$\varepsilon \phi_{KP}{}^{\infty}[\ldots \phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R \mid_{1, l'_1}) \mid_{2, l'_2}] \ldots \mid_{M, l'_M}.$$

Therefore, a search path must exist and $(u_1, l_1, \ldots, u_N, l_N)$ belongs to the fixed point with respect to the path. However, since the path results in a single valued non-empty set, a consistent labeling is found which contains $(u_1, l_1, \ldots, u_N, l_N)$.

Q.E.D.

Lemma 2.6: If $(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha_1/\beta_1, \ldots, a_M/\beta_M)$ and there exists a j, $1 \le j \le M$, such that $L = \beta_j$, then $(u_1, l_1, \ldots, u_N, l_N)$ does not contribute to any consistent labeling.

Proof: Suppose $(u_1, l_1, \ldots, u_N, l_N)$ is involved with a consistent labeling, then there must exist a search path which leads to a single-valued set and $(u_1, l_1, \ldots, u_N, l_N)$ belongs to the set. But this cannot be true in this case, according to **Definition 3.**

Q.E.D.

It turns out that the states can be used to record the result of the tree search. For example, it is possible to assume that the tree search starts with every N-tuple of pairs in state $(\phi/\phi, \ldots, \phi/\phi)$. If a search leads $(u_1, l_1, \ldots, u_N, l_N) \varepsilon \phi_{KP}{}^{\infty}[\ldots \phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R \mid_{u'_1, l'_1}) \mid_{u'_2, l'_2}] \ldots \mid_{u'_t, l'_t}]$, then record it as $(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha_1/\phi, \ldots, \alpha_M/\phi)$ where $\alpha_{u'_i} = \{l'_i\}$. If $(u_1, l_1, \ldots, u_N, l_N) \varepsilon \phi_{KP}{}^{\infty}[\ldots \phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R \mid_{u'_1, l'_1}) \mid_{u'_2, l'_2}] \ldots \mid_{u'_t, l'_t}$ but $(u_1, l_1, \ldots, u_N, l_N) \notin \phi_{KP}{}^{\infty}[\ldots \phi_{KP}{}^{\infty}[\phi_{KP}{}^{\infty}(R \mid_{u'_1, l'_1}) \mid_{u'_2, l'_2}, \ldots \mid_{u'_{t-1}, l'_{t-1}}]$, then record it as $(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\phi/\beta_1, \ldots, \phi/\beta_M)$ where $\beta'_{t+1} = \{l'_{t+1}\}$, etc. Furthermore, Lemma 2.4 provides a way to combine some individual states into an integrated state.

To be specific, the following concept is provided:[2]

Definition 4: A state $(\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)$ is a most general state of $(u_1, l_1, \ldots, u_N, l_N)$ with respect to ϕ_{KP} if and only if the following conditions are satisfied:

1). $(u_1, l_1, \ldots, u_N, l_N)_{\phi_{KP}} \rightarrow (\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)$.

2). For any free unit i (e.g. $i \notin \{u_1, \ldots, u_N\}$), $i = 1, \ldots, M$, $L = \alpha_i \cup \beta_i$ and $\alpha_i \cap \beta_i = \phi$ is satisfied.

The consistent labeling problem can be restated in terms of relation-state pair as follows:

Definition 5: A book-keeping (R, T)-consistent model is a quadruple $((U, L, T, R), \phi_{KP}, S, F)$, where

(U, L, T, R) is a (T,R)-consistent model;

ϕ_{KP} is a look-ahead operator;

S is the state space of R;

F is a mapping from one relation-state pair to another relation-state pair which is defined as follows:

For any $R' \subseteq R$, $1 \le i \le M$, $l \varepsilon L$, then

F: $<(u_1, l_1, \ldots, u_N, l_N), (\alpha_1/\beta_1, \ldots, \alpha_M/\beta_M)> \rightarrow$
$<(u_1, l_1, \ldots, u_N, l_N), (\alpha'_1/\beta'_1, \ldots, \alpha'_M/\beta'_M)>$

where

[2] Without loss of generality, assume that any N-tuple of pairs in R participates at least one consistent labeling. This assumption makes it possible to dispense with some trivial special cases.

$\alpha'_i=\alpha_i\cup\{l\}$, $\beta'_i=\beta_i$, $\alpha'_j=\alpha_j$, $\beta'_j=\beta_j$, $j\neq i$.
If $(u_1,l_1,\ldots,u_N,l_N)\varepsilon\phi_{KP}{}^{\infty}(R'|_{i,l})$
$\beta'_i=\beta_i\cup\{l\}$, $\alpha'_i=\alpha_i$, $\alpha'_j=\alpha_j$, $\beta'_j=\beta_j$, $j\neq i$.
If $(u_1,l_1,\ldots,u_N,l_N) \notin \phi_{KP}{}^{\infty}(R'|_{i,l})$ and $(u_1,l_1,\ldots,u_N,l_N)\varepsilon R'|_{i,l}$
$\alpha'_1=\alpha_1$, $\beta'_1=\beta_1$,
otherwise.

The consistent labeling problem is:

Given an initial relation-state pair set:

$S_I = \{<(u_1,l_1,\ldots,u_N,l_N),(\phi/\phi,\ldots,\phi/\phi)>|(u_1,l_1,\ldots,u_N,l_N)\varepsilon R\}$,

find the final relation-state pair set:

$S_F = \{<(u_1,l_1,\ldots,u_N,l_N),(\alpha_1/\beta_1,\ldots,\alpha_M/\beta_M)>|(u_1,l_1,\ldots,u_N,l_N)\varepsilon R$ such that $(\alpha_1/\beta_1,\ldots,\alpha_M/\beta_M)$ is the most general state of $(u_1,l_1,\ldots,u_N,l_N)\}$.

The book-keeping model can be tailored gracefully to the tree search. Suppose the tree search start at level 0, e.g. $R_0=R$, then the following procedure is proposed:

1). Go to the next level: $R_i=R_{i-1}|_{i,l'_i}$ where l'_i is the next legal label available.

2). State transition: suppose the current state of $(u_1,l_1,\ldots,u_N,l_N)$ is $(\alpha_1/\beta_1,\ldots,\alpha_M/\beta_M)$ where $(u_1,l_1,\ldots,u_N,l_N)\varepsilon R_i$. If $l'_i\varepsilon\alpha_i\cup\beta_i$ then the state $(u_1,l_1,\ldots,u_N,l_N)$ is unchanged.
Otherwise, if $(u_1,l_1,\ldots,u_N,l_N)\varepsilon\phi_{KP}{}^{\infty}R_i$, then the state of $(u_1,l_1,\ldots,u_N,l_N)$ becomes $(\alpha_1/\beta_1,\ldots,(\alpha_i\cup\{l'_i\})/\beta_i,\ldots,\alpha_M/\beta_M)$ else it becomes $(\alpha_1/\beta_1,\ldots,\alpha_i/(\beta_i\cup\{l'_i\}),\ldots,\alpha_M/\beta_M)$.

3). Check for consistency:

Case 1:
$\phi_{KP}{}^{\infty}R_i=\phi$. $i:=i-1$, goto 1.

Case 2:
$\phi_{KP}{}^{\infty}R_i$ is single-valued. A consistent labeling, say $(\bar{l}_1,\ldots,\bar{l}_M)$, is found. For each relation-state pair $<(u_1,l_1,\ldots,u_N,l_N)$, $(\alpha_1/\beta_1,\ldots,\alpha_M/\beta_M)>$ such that $(u_1,l_1,\ldots,u_N,l_N)\varepsilon\phi_{KP}{}^{\infty}R_i$ update $\alpha_{i+1},\ldots,\alpha_M$ properly (e.g. $\alpha_j:=\alpha_j\cup\{\bar{l}_j\}$, $j=i,\ldots,M\}$. $i=i-1$, goto 1.

Case 3:
Otherwise. $R_i=\phi_{KP}{}^{\infty}R_i$, $i=i+1$, goto 1.

The following demonstrates how a book-keeping method can be used for the solution of this example:

Step 1: Fixing 1 to a yields Figure 2. Since $\phi_{KP}{}^{\infty}R_1=R_1$. Every relation in Figure 2 has a same updated state ({a}/ϕ,ϕ/ϕ,ϕ/ϕ,ϕ/ϕ,ϕ/ϕ).

Step 2: Fixing 2 to a yields Figure 3. The fixed point of Figure 3 is calculated to be Figure 4. Since Figure 4 is single-valued, a consistent labeling (a,a,b,c,d) has been found. Furthermore, the states of the relations in Figure 3 have been updated as follows: (1,a,2,a), (1,a,3,b), (1,a,4,c), (1,a,5,d), (2,a,3,b), (2,a,4,c), (2,a,5,d), (3,b,4,c), (3,b,5,d), and (4,c,5,d) have the updated state ({a}/ϕ,{a}/ϕ,{b}/ϕ,{b}/ϕ,{c}/ϕ,{d}/ϕ) while (1,a,3,a), (1,a,3,c), (1,a,4,b), (1,a,5,a), (3,a,4,b), (3,a,5,a), (3,c,4,c), and (3,c,5,a) have the updated state ({a}/ϕ,ϕ/{a},ϕ/ϕ,ϕ/ϕ,ϕ/ϕ).

Step 3: Backtrack one level. Fixing 1 to d yields Figure 5. the fixed point of Figure 5 is calculated to be Figure 6. Since Figure 6 is single-valued, a consistent labeling (a,d,a,b,a) has been found. Furthermore, the states of the relations in Figure 5 have been updated as follows: (1,a,2,d), (2,d,3,a), (2,d,4,b), and (2,d,5,a) have the updated state ({a}/ϕ,{d}/ϕ,{a}/ϕ,{b}/ϕ,{a}/ϕ); (1,a,3,a), (1,a,4,b), (1,a,5,a), (3,a,4,b), (3,a,5,a), and (4,b,5,a) have the updated state ({a}/ϕ,{d}/{a},{a}/ϕ,{b}/ϕ,{a}/ϕ); (1,a,3,c), (3,c,4,c), (3,c,5,a), and (3,c,5,d) have the updated state ({a}/ϕ,ϕ/{a,d},ϕ/ϕ,ϕ/ϕ,ϕ/ϕ); (1,a,3,b), (1,a,4,c), (1,a,5,d), (3,b,4,c), (3,b,5,d), and (4,c,5,d) have the state ({a}/ϕ,{a}/{d},{b}/ϕ,{c}/ϕ,{d}/ϕ); (2,d,3,c) and (2,d,4,c) have the updated state ({a}/ϕ,ϕ/{d}, ϕ/ϕ,ϕ/ϕ,ϕ/ϕ), etc.

At this moment, (1,a,3,c), (3,c,4,c), (3,c,5,a), and (3,c,5,d) have been shown that they cannot contribute to any consistent labeling, by **Lemma 2.4**. Thus, they can be removed from R. Moreover, (2,d,3,c) should also be removed from R, since unit 2 is fixed to d and d $\varepsilon\beta_2$.

Step 4: Back up two levels. Fixing 1 to b yields Figure 7. The fixed point of Figure 7 is calculated to be Figure 8.

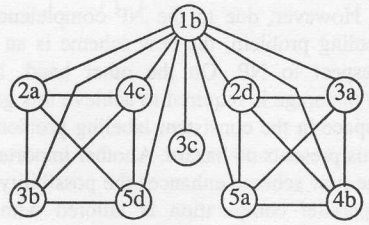

Figure 7. R_1 set.

The states of the relations in Figure 7 have been updated as follows: (1,b,3,c) has the updated state (ϕ/{b},ϕ/ϕ,ϕ/ϕ,ϕ/ϕ,ϕ/ϕ); (1,b,2,a), (1,b,2,d), (1,b,3,a), (1,b,3,b), (1,b,4,b), (1,b,4,c), (1,b,5,a), and (1,b,5,d) have the updated state

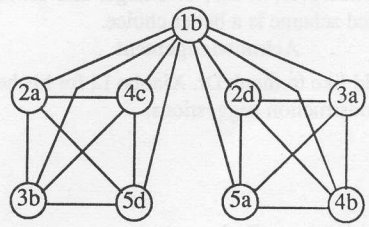

Figure 8. $\phi_{13}{}^{\infty}R_1$ set.

({b}/ϕ,ϕ/ϕ,ϕ/ϕ,ϕ/ϕ,ϕ/ϕ);

(2,d,3,a), (2,d,4,b), and (2,d,5,a) have the updated state ({a,b}/ϕ,{d}/ϕ,{a}/ϕ,{b}/ϕ,{a}/ϕ); (3,a,4,b), (3,a,5,a), and (4,b,5,a) have the updated state ({a,b}/ϕ,{d}/{a},{a}/ϕ,{b}/ϕ,{a}/ϕ); (2,a,3,b), (2,a,4,c), and (2,a,5,d) have the updated state ({a,b}/ϕ,{a}/ϕ,{b}/ϕ,{c}/ϕ,{d}/ϕ); (3,b,4,c), (3,b,5,d), and (4,c,5,d) have the updated state ({a,b}/ϕ,{a}/{d}, {b}/ϕ,{c}/ϕ,{d}/ϕ).

Again, (1,b,3,c) can be discarded. The next consistent labeling (b,a,b,c,d) is obtained by fixing 2 to a. This time the application of ϕ_{KP} to (2,a,4,c), (2,a,5,d), (2,a,3,b), (3,b,4,c), (3,b,5,d), and (4,c,5,d) is avoided. Similarly, when fix 2 to d, the computation of (2,d,3,a), (2,d,4,b), (2,d,5,a), (3,a,4,b), (3,a,5,a), and (4,b,5,a) can be saved.

3. Conclusions

In this paper, a new book-keeping approach to the labeling problem, based on the Independent computation theorem, is developed. The proposed scheme takes advantage of the computation as it progresses, using a state table for each N-tuple of pairs in the relation set. As the result of the new scheme, unnecessary repetitive computation is avoided. It has been proved that in the worst case the time complexity is reduced by a factor of polynomial $\binom{N}{K}\binom{M}{P-K}\#L^{P-K-1}\#T$, see Appendix A. However, due to the NP completeness of the consistent labeling problem, the new scheme is an improvement with respect to NP. On the other hand, a cost of $\#R\,(N+\#L\,(M-N))$ storage is required to achieve this goal. Since the memory space in the consistent labeling problem is not a bottle neck, this presents no hazard. Another important advantage is that the new scheme enhances the possibility of parallelism. The parallel computation is tailored with the tree search in the previous approach in a sense that several processors may work in parallel to explore different branches. With the new scheme, however, the search can start at any unit in any order. More importantly, since the results of the search are summarized in the form of states, each processor shares the result with others. Finally, although the proposed scheme outperforms previous schemes in the worst case, the chance of the worst case happening in practical data is expected to be rare. Thus for a simulation based on real data is encouraging. It is believed, however, in case of a larger and denser relation set the proposed scheme is a better choice.

Acknowledgement

We would like to thank Dr. Xiaobo Li for his helpful discussion and construction suggestions.

References

[1]. D.H. Ballard and C.H. Brown "Computer Vision", Prentice-Hall, Inc. Englewood Cliffs, NJ, 1982.

[2]. M.R.Garey and D.S. Johnson, "Computers and Intractability", W.H Freeman and Company, NY, 1979.

[3]. A. Ginzberg, Algebraic "Theory of Automata", New YorK: Academic, 1976.

[4]. R.M Haralick, L.S Davis and A. Rosenfeld, "Reduction Operation for Constraint Satisfaction", Information Science 14, 1978, 199-219.

[5]. R.M Haralick and L.G Shapiro,"The Consistent Labeling Problem: Part I", IEEE Trans. Pattern Anal. Machine Intell. Vol PAMI-1, pp. 173-184. Apr. 1979.

[6]. R.M Haralick and L.G Shapiro,"The Consistent Labeling Problem: Part II", IEEE Trans. Pattern Anal. Machine Intell. Vol PAMI-2, pp. 193-203. May. 1980.

[7]. A. Rosenfeld, R.A. Hummel and S.W Zucker, "Scene Labeling by Relaxation Operations", IEEE, Trans. Syst. Man. Cybern. Vol. SMC-6. pp.420-433, June, 1976.

[8]. A. Rosenfeld and A.Kak, "Digital Picture Processing", New York: Academic, 1976.

Appendix A

Time and Space Complexity

To verify the feasibility of the new scheme, a comparison of the worst case time complexity of the operation involved in computing one N-tuple of pairs between previous approach and new approach is made. In addition, the memory requirement for the new scheme is given.

Time Complexity

Lemma A1: The time complexity of the computation involved with an N-tuple of pairs, $(u_1,l_1,\ldots,u_N,l_N)$, using the previous approach is bounded by $O(b\,\mu\lambda)$, where b is the number of branches in the search tree, $\mu=M+1+\#R$ is the maximum number of iterations applying ϕ_{KP} to a branch, $\lambda=\binom{N}{K}\binom{M}{P-K}\#L^{P-K}\#T$ is the number of operation involved during each iteration of ϕ_{KP} with respect to $(u_1,l_1,\ldots,u_N,l_N)$.

Proof: See [6].

Lemma A2: The time complexity of the computation involved with $(u_1,l_1,\ldots,u_N,l_N)$, using the book-keeping method is $O(b\,\mu\#L)$.

Proof: The only difference between the book-keeping method and the previous method is that when $(u_1,l_1,\ldots,u_N,l_N)$ is chosen for the test of consistency, the process first checks whether a similar test has been performed before. If so no further operation is needed, otherwise ϕ_{KP} is applied.

Since each unit has $\#L$ possible labels, the check of the state of $(u_1,l_1,\ldots,u_N,l_N)$ cannot take more than $\#L$ steps. On the other hand, since each time ϕ_{KP} is applied to $(u_1,l_1,\ldots,u_N,l_N)$ a new label is assigned to its state table, and the size of the table is bounded by $N+\#L\,(M-N)$, the number of iterations of applying ϕ_{KP} to $(u_1,l_1,\ldots,u_N,l_N)$ is bounded by $O(N+\#L\,(M-N))$.

Therefore, the total time is:

$$O(b\,\mu\#L+(N+\#L\,(M-N)\lambda)).$$

Since in the worst case b is an exponential of M, the second term can be ignored, hence, the Lemma is proved.

Q.E.D.

Space Complexity

The book-keeping nature of the method decides that it is essentially a compromise between time and space. An naive implementation associates every N-tuple of pairs with a state table, each of which requires $N+\#L(M-N)$ memory units. Therefore, a total of $\#R(N+\#L(M-N))$ extra storage is necessary. If, however, labels are assigned to units in a specific order, then approximately half of spaces can be saved since only α (or β) states need to be recorded. An even better bound can be achieved if a new data structure is introduced.

Despite of the extra memory requirement, the new scheme is still considered to be feasible. The main justifications are:

1). In the consistent labeling problem, the memory requirement for the algorithm is low, thus the extra memory is expected to be affordable.

2). The amount of computation saved by the new scheme in the worst case is very attractive.

Finally, it should be pointed out that the process of calculating $\phi_{KP}{}^{\sim}R$ is just the labeling problem all over again with P units instead of M and a restricted relation instead of R. Thus, $\phi_{KP}{}^{\sim}R$ can be implemented repeatedly by tree search, using ϕ_{KP^*} with $P^*<P$ [5]. This implies that the book-keeping method can be used repeatly. If so the space requirement will be increased accordingly (it will be still a polynomial of M, however). Since for the small sub-problem case, the book-keeping method may not be a good choice, heuristic rules can be applied to achieve even better performance.