Monotone conjunctive structures of second order

M.I. Schlesinger,[1] *B. Flach*[2]

[1]International Research and Training Centre of Information Technologies and Systems,
National Academy of Science of Ukraine. Email: *schles@image.kiev.ua*
[2]Institute of Artificial Intelligence, Dresden University of Technology. Email: *bflach@ics.inf.tu-dresden.de*

Abstract

We introduce a class of conjunctive structures of second order for which the consistent labeling problem is solvable in polynomial time. This result is generalized then to a class of "maxmin" problems.

1. Introduction

The "consistent labeling" or "exact matching" problem is usually introduced in following terms: An undirected graph G is given with finite sets of possible labels for each node. Furthermore for each edge of G constraints on label pairs are given . Such a structure is called consistent if there exist allowable labelings of the nodes, i.e. labelings which satisfy all constraints. A well known example is the colorability problem: A graph G is k-colorable if there exists an assignment of the integers $1,2,\ldots,k$, called "colors" to the nodes such that no two adjacent nodes are assigned the same color. Another well known example is the 2-CNF satisfiability problem. A boolean expression is said to be in 2-conjunctive normal form (2-CNF) if it is the product of sums of at most 2 literals. It is satisfiable if some assignement of 0's and 1's to the variables gives the expression the value 1 (it is leaved to the reader to transform this in a consistent labeling problem). Though we supposed here that constraints are given only on pairs of labels (i.e. on the edges of G) it is of course possible to extend the consistent labeling problem considering constraints on k-tuples of nodes. An example is the k-CNF satisfiability problem.

The necessity to have efficient algorithms for consistent labeling arise from various fields in computer science. Particularly we are interested in structural image recognition especially using two dimensional image grammars. Suppose we consider images on a rectangular pixel lattice with values from a finite alphabet. The aim is then to describe subsets of images by means of two dimensional grammars analogous to the description of string subsets by means of formal (one dimensional) languages. Loosely speaking we can imagine a regular image grammar in the following way: The pixel values are called terminal symbols and a second alphabet of nonterminal symbols is introduced which serve as "interpretations" of local fragments in the terminal image: We assign a subset of nonterminals to each local fragment occuring in images of the desired set. These are the possible interpretations of the fragment. Then constraints are introduced on adjacent (e.g. horizontal and vertical) nonterminal symbol pairs. Testing if a given image belongs to the grammar language leads to a consistent labeling problem (in this context often called "exact matching"): For every fragment of the given image we obtain a subset of possible nonterminal symbols and have to assign one of these symbols to the fragment so that all constraints are fulfilled (see the example below). Of course this is only the first and "easiest" problem in structural image recognition. A direct generalization is the "best matching" problem: Given a grey valued image we obtain only weights for each possible terminal symbol in each pixel (and not a unique terminal symbol as before). The aim is then to find the symbol image belonging to the grammar language which maximizes some objective function e.g. the sum of weights or the minimal occuring weight.

Despite the importance of the consistent labeling problem its long history (e.g. [7, 2, 4, 1, 5, 3]) lacks attempts to find relevant subclasses of the problem which can be solved in polynomial time.[1] The stochastic relaxation labeling introduced by A. Rosenfeld also doesn't clarify the problem. Previously one of us (M.I.S.) investigated how this problem does simplify in the case of not fully connected graphs [6]. An algorithm was found, whose complexity depends on the complexity of the graph G.[2]

In this paper we propose another approach for tackling the problem: namely by restricting the class of used predicates. We show how the exact matching problem and the "maxmin" problem (a generalization thereof) can be solved in polynomial time for a class of conjunctive structures.

2. Monotone conjunctive structures

Let G be a complete graph with nodes denoted by $r \in G$. A finite set $S(r)$ of symbols is attached to each node $r \in G$. Furthermore a predicate $\chi_{rr'}$ on $S(r) \times S(r')$ is given for every pair (r,r') of nodes (more precisely: for every edge of G). We call such triple $\mathcal{K} = (G,S,\chi)$ a conjunctive

[1]In general the exact matching problem is NP-complete.

[2]Unfortunately the algorithm also scales exponentially for complex graphs.

structure of second order. A symbol field $\mathbf{s}$ on $\mathcal{K}$ assigns a symbol $\mathbf{s}(r) \in S(r)$ to each node $r \in G$. A symbol field $\mathbf{s}$ is said to be allowable if

$$\chi_{rr'}\left(\mathbf{s}(r), \mathbf{s}(r')\right) = 1$$

for every edge of G. A structure $\mathcal{K}$ is said to be consistent if there exists at least one allowable symbol field on $\mathcal{K}$. If $\mathcal{K}'$ and $\mathcal{K}$ are conjunctive structures, then $\mathcal{K}'$ is called a substructure of $\mathcal{K}$ if and only if $G' = G$, $\forall r \in G$ $S'(r) \subset S(r)$ and the predicates fulfil

$$\chi'_{rr'} \leq \chi_{rr'}|_{S'(r) \times S'(r')} .$$

In case of equality $\mathcal{K}'$ is called induced substructure of $\mathcal{K}$. The kernel $\mathrm{Ker}(\mathcal{K})$ of a conjunctive structure is defined as the smallest substructure possessing the same allowable symbol fields as $\mathcal{K}$.

The "exact matching" or "consistent labeling" problem implies to prove the consistence of a given conjunctive structure or its substructures.

In the remainder we suppose that symbol sets $S(r)$ are *ordered* in structures under consideration. A subset I of an ordered set S is called interval if

$$I = \{s \in S \mid s_1 \leq s \leq s_2\}$$

with some s_1, $s_2 \in S$. We write $I_1 \leq I_2$ if corresponding inequalities hold for the boundaries of the intervals.

A conjunctive structure $\mathcal{K} = (G, S, \chi)$ is called *strong monotone* if 1–3 hold:

1. Every set $S(r)$ is ordered.
2. The sets
$$S(r|r', s') = \{s \in S(r) \mid \chi_{rr'}(s, s') = 1\}$$
are nonempty intervals in $S(r)$.
3. The boundaries of the intervals $S(r|r', s')$ are monotonously increasing functions of s' i.e. $S(r|r', s'_2) \geq S(r|r', s'_1)$ if $s'_2 \geq s'_1$.

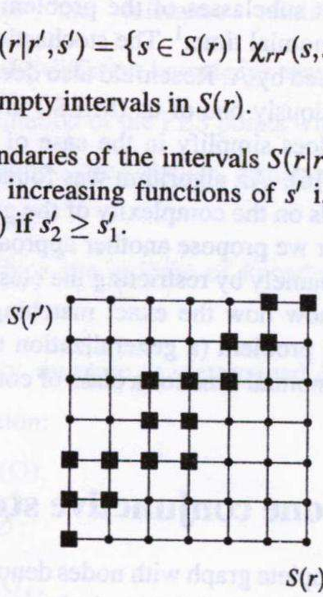

Figure 1: Example predicate of a monotone structure

Remark 1 If $s'_1, s'_2 \in S(r')$ are consecutive in $S(r')$ then from 2 and 3 follows that the union of the intervals $S(r|r', s'_1) \cup S(r|r', s'_2)$ is again an interval in $S(r)$.

We call a conjunctive structure *monotone* if the property 2 is weakened in so far as the intervals could be empty and 3 holds only for nonempty intervals. It is easy to see that the predicates of a monotone structure could be written as

$$\tilde{\chi}_{rr'}(s_1, s_2) = g_1(s_1)\chi_{rr'}(s_1, s_2)g_2(s_2)$$

where χ is a predicate of a strong monotone structure and g_1, g_2 are arbitrary binary functions.

Remark 2 It is easy to prove that an induced substructure of a monotone structure is monotone.

Suppose $\mathcal{K}$ is a conjunctive structure, (r, r', r'') a triple of nodes and $s \in S(r)$, $s' \in S(r')$ a pair of symbols with $\chi_{rr'}(s, s') = 1$. We say that there is a cycle through this pair on the triangle (r, r', r'') if there exists at least one $s'' \in S(r'')$ such that

$$\chi_{rr''}(s, s'') = 1 \quad \text{and} \quad \chi_{r'r''}(s', s'') = 1 .$$

We call a structure *cyclic* if this holds for every allowable pair of symbols on every triangle.

Lemma 1 Let $\mathcal{K}$ be a strong monotone and cyclic conjunctive structure, $G_0 \subset G$ an induced subgraph and $\mathbf{s}_0$ an allowable symbol field on G_0. Then there exist at least one allowable extension of $\mathbf{s}_0$ on $\mathcal{K}$.

Proof: We prove this by induction over size of the remaining graph. Let r be a node not contained in G_0. We consider the sets $S(r|r', \mathbf{s}_0(r'))$ for all $r' \in G_0$. They are intervals in $S(r)$. If their intersection is nonempty, the symbol field can be extended on r. Suppose in contrary that their intersection is empty. Then there exists a pair of nodes $r_1, r_2 \in G_0$ so that

$$S(r|r_1, \mathbf{s}_0(r_1)) \cap S(r|r_2, \mathbf{s}_0(r_2)) = \emptyset .$$

But this contradicts the assumption: there must be at least one cycle through $\mathbf{s}_0(r_1)$, $\mathbf{s}_0(r_2)$ on the triangle (r_1, r_2, r). $\square$

Let $\mathcal{K} = (G, S, \chi)$ be a monotone structure whose consistence is to be proved. Suppose $\mathcal{K}$ is consistent. If there exists an edge (r, r') and a symbol $s \in S(r)$ such that $S(r'|r, s)$ is empty, none of the allowable symbol fields can have value s in r. Therefore all allowable symbol fields of $\mathcal{K}$ are also allowable symbol fields of the induced substructure with $S'(r) = S(r) \setminus s$. We call this restriction of the symbol set "removing of a symbol with no incident symbol edges". Suppose furthermore that there exists a triple (r, r', r'') of nodes and a pair of symbols $s_0 \in S(r)$, $s'_0 \in S(r')$ and $\chi_{rr'}(s_0, s'_0) = 1$ with no cycle through s_0, s'_0 on (r, r', r''). Then all allowable symbol fields of $\mathcal{K}$ are also allowable symbol fields of the substructure with

$$\chi'_{rr'}(s, s') = \begin{cases} 0 & \text{if } s = s_0 \text{ and } s' = s'_0, \\ \chi_{rr'}(s, s') & \text{else.} \end{cases}$$

We call this restriction of the predicate "removing of noncyclic symbol edges". It is clear that both operations preserve the kernel and therefore can be applied iterative.

The algorithm for proving the consistence repeatedly applies the following three steps:

1. Iterative removing of symbols with no incident symbol edges. The obtained induced substructures of $\mathcal{K}$ are selves monotone (see Rem. 2). This removing stops if the obtained induced substructure $\mathcal{K}'$ becomes strong monotone.

2. Choose a node $r_0 \in G$. For every edge (r, r') with $r, r' \in G_1 = G \setminus r_0$ remove all symbol edges which are noncyclic on the triangle (r_0, r, r'). We obtain a substructure $\mathcal{K}'' \subset \mathcal{K}'$.
3. Remove node r_0 i.e. restrict the structure $\mathcal{K}''$ on G_1. We denote the resulting structure $\mathcal{K}_1$.

This algorithm stops either if a symbol set becomes empty or if the resulting graph has one node and a nonempty symbol set on it.

Theorem 1 Let $\mathcal{K} = (G, S, \chi)$ be a monotone structure. The algorithm stops with a one node graph and a nonempty symbol set if and only if $\mathcal{K}$ is consistent.
Proof: In order to prove this theorem we show that $\mathcal{K}_1$ is monotone and that each allowable symbol field on $\mathcal{K}_1$ can be extended on $\mathcal{K}$. (The converse is true because it is evident that $\mathcal{K}''$ contains $\mathrm{Ker}(\mathcal{K})$.) So let us first show that $\mathcal{K}''$ is monotone. The structure $\mathcal{K}''$ yields from the strong monotone structure $\mathcal{K}'$ by removing noncyclic symbol edges. Let (r_0, r, r') be a triple of nodes. Consider the sets

$$S(r' | r, s; r_0) = \bigcup_{s_0 \in S(r_0 | r, s)} S(r' | r_0, s_0)$$

for every $s \in S(r)$. It is easy to see that they are intervals in $S(r')$ (see Remark 1). Moreover: their boundaries are monotonous increasing functions of s. This holds also for the intersections

$$S(r' | r, s; r_0) \cap S(r' | r, s)$$

if they are nonempty. Therefore $\mathcal{K}''$ is monotone. Symbol edges on the edges (r_0, r) are not removed during the transition from $\mathcal{K}'$ to $\mathcal{K}''$. So the proof of Lemma 1 can be repeated here to show that each symbol field on $\mathcal{K}_1$ can be extended on $\mathcal{K}''$. $\square$

Example 1 In this example we show a monotone structure arising from a simple image grammar. Consider images on a rectangular pixel lattice with values black and white. Suppose the image set under consideration consists of all images containing a "horizontal" black curve of thickness 1 pixel. Horizontal means that every column of the image contains at most one black pixel. Curve means

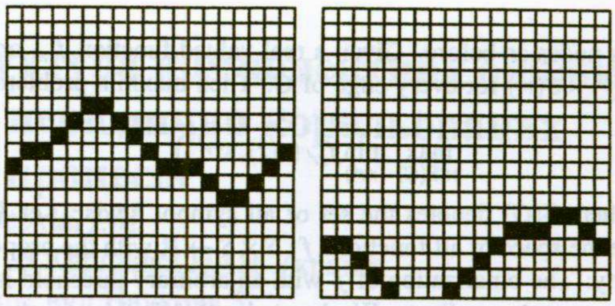

Figure 2: Images belonging to the grammar language

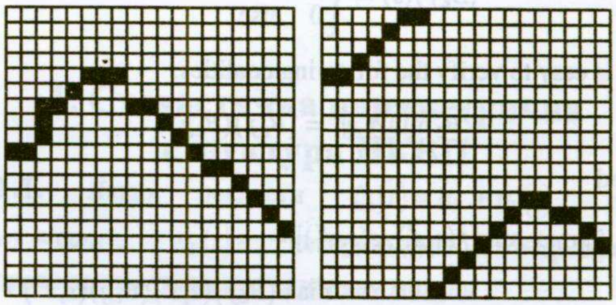

Figure 3: Images not belonging to the grammar language

that black pixels are connected in the 8-neighbourhood. A simple grammar describing this set of images can be constructed as follows: The alphabet of terminal symbols is $\{b, w\}$ which stand for black and white. The alphabet of nonterminal symbols is $\{B, C, A\}$ which stand for "below", "curve" and "above". The fragments of the terminal image are the pixels itself. A pixel with terminal symbol (w)hite can be assigned to the nonterminal symbols B, A. A pixel with terminal symbol (b)lack–only to the nonterminal symbol C. For horizontal adjacent pixels the allowed symbol pairs are

$$\{(B,B), (B,C), (C,B), (C,C), (C,A), (A,C), (A,A)\}$$

i.e. there are only two forbidden pairs: (B, A) and (A, B). For vertical adjacent pixels the allowed symbol pairs are

$$\left\{ \begin{pmatrix} B \\ B \end{pmatrix}, \begin{pmatrix} C \\ B \end{pmatrix}, \begin{pmatrix} A \\ C \end{pmatrix}, \begin{pmatrix} A \\ A \end{pmatrix} \right\}.$$

For every other pixel pair all symbol pairs are allowed. Taking now the order $\{B, C, A\}$ it is easy to see that the corresponding structure is monotone. Therefore the exact matching problem for this grammar is solvable in polynomial time. $\square$

3. The maxmin problem

In this section we consider the "maxmin" problem–a generalization of the "exact matching" problem. Let G be a complete graph with symbol sets $S(r)$ attached to each of its nodes as before. Given a real valued function $f_{rr'}$ on $S(r) \times S(r')$ for every edge of G. Then maxmin problem is

$$\max_{s \in S(G)} \min_{(r,r')} f_{rr'}(s(r), s(r'))$$

where $S(G)$ denotes the set of all symbol fields. Let $\mathfrak{F}$ be the space of all functions $f : S \times S \to \mathbb{R}$ with the property: The binarization of f with an arbitrary threshold is a monotone predicate. We denote by $B_\varepsilon f$ the binarization with treshold ε:

$$(B_\varepsilon f)(s) = \begin{cases} 1 & \text{if } f(s) > \varepsilon, \\ 0 & \text{else.} \end{cases}$$

It is easy to verify the following eqalities:

$$B_\varepsilon \min[f, g] = (B_\varepsilon f)(B_\varepsilon g)$$

and

$$B_\varepsilon \max_s \min[f(s, s'), g(s, s'')] = \max_s (B_\varepsilon f)(s, s')(B_\varepsilon g)(s, s'')$$

Since we have shown that the space of monotone predicates is closed under the operations in the right hand sides of the above equations, it follows that so is $\mathfrak{F}$ under the operations in the left hand sides.

We suppose that all weight functions $f_{rr'}$ of the maxmin problem under consideration are in $\mathfrak{F}$. Let r_0 be a node of G and $G' = G \setminus r_0$. The maxmin problem can be transformed into

$$\max_{s' \in S(G')} \min\left[\max_{s_0} F_0(s_0, s'), F_{G'}(s')\right]$$

with

$$F_0(s_0, s') = \min_{r \in G'} f_{r_0 r}(s_0, s'(r))$$

$$F_{G'}(s') = \min_{(r,r') \in G' \times G'} f_{rr'}(s'(r), s'(r')) .$$

Furthermore

$$\max_{s_0} F_0(s_0, s') = \max_{s_0} \min_{r \in G'} f_{r_0 r}(s_0, s'(r))$$

$$= \min_{(r,r') \in G' \times G'} \max_{s_0} \min\left[f_{r_0 r}(s_0, s'(r)), f_{r_0 r'}(s_0, s'(r'))\right]$$

$$= \min_{(r,r') \in G' \times G'} h_{rr'}(s'(r), s'(r'))$$

holds because the sets $\{s \in S \mid f(s, s') > \varepsilon\}$ are intervals in S. The functions $h_{rr'}$ are elements of $\mathfrak{F}$ and finally we obtain

$$\max_{s \in S(G)} \min_{(r,r') \in G \times G} f_{rr'}(s(r), s(r')) = \max_{s' \in S(G')} \min_{(r,r') \in G' \times G'} \hat{f}_{rr'}(s(r), s(r'))$$

with $\hat{f}_{rr'} = \min[f_{rr'}, h_{rr'}]$ which are elements of $\mathfrak{F}$. We have shown therefore how to reduce the maxmin problem on G to an equivalent problem on $G' = G \setminus r_0$. By repeating this step we can solve the problem in polynomial time.

Conclusion

We have shown that "consistent labeling" and "maxmin" problems can be solved in polynomial time for monotone conjunctive structures of second order. Although the algorithms presented here are sequential they can be reformulated easily in a parallel distributed manner. The parallel versions generate in addition the kernel for the consistent labeling problem and the best symbol field for the maxmin problem respectively.

References

[1] L.S. Davis and A. Rosenfeld. Cooperating processes for low-level vision: A survey. In R. Kasturi and R. Jain, editors, *Computer Vision: Principles*, pages 282–299. IEEE Computer Soc. Press, 1991.

[2] R.A. Hummel and S.W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. PAMI*, 5:267–286, 1983.

[3] W.K. Koval and M.I. Schlesinger. Two dimensional programming in image analysis problems. *Automation and Telemechanics*, (8):149–168, 1976.

[4] A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. SMC*, 6:420–433, 1976.

[5] M.I. Schlesinger. Syntactic analysis of disturbed two dimensional visual signals. *Cybernetics*, (4):113–130, 1976.

[6] M.I. Schlesinger. *Mathematical Methods of Image Processing*. Naukova Dumka, Kiev, 1989.

[7] D.L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Technical Report A1271, MIT, 1972.