相接行凸约束网络的快速识别算法*

陈恩红, 张振亚, 王煦法

(中国科学技术大学 计算机科学与技术系,安徽 合肥 230026)

E-mail: cheneh@ustc.edu.cn

http://mail.ustc.edu.cn/~zzychm

摘要: 约束网络为计算机科学中的许多问题提供了一种有效的表示方法.一般而言,约束满足问题是 NP 完全的.然而,许多实际问题通常对约束的结构或形式施加了特殊的限制,从而能够高效地加以解决.迄今,为了识别易处理约束类,人们对特殊的约束或约束网络方面进行了许多研究.相接行凸(connected row-convex,简称 CRC)约束网络是 Deville 等人提出的一类易处理问题.为了给该类问题寻求有效的快速识别算法,在 CRC 约束网络相关工作基础上,提出了 CRC 约束矩阵的标准型.在分析 CRC 约束矩阵的标准型性质的基础上,利用行凸(row-convex,简称 RC)约束网络的判定,结合 PQ 树(由 P 节点和 Q 节点构成的树)的性质和矩阵的索引表示法,给出了 CRC 约束网络的快速识别算法.该算法的时间复杂度为 $O(n^3d^2)$,其中,n 为约束网络涉及的变量数,d 为各变量的定义域中最大定义域的大小.该时间复杂度达到该类问题的最佳时间复杂度,从而为实际的 CRC 约束满足问题的求解提供了可行的方法.

关 键 词: 二项约束网络;RC 约束网络;CRC 约束网络;PQ 树;CRC 约束矩阵的标准型

中图法分类号: TP301 文献标识码: A

约束满足问题(constraint satisfaction problem,简称 CSP)可以描述为:(1) 存在一个有限集 S,S 中的元素称作变量;(2) S 中的任一变量存在一个值域;(3) 对 S 中的变量存在一组约束关系.CSP 的一个解是指 S 中的每一个变量在其值域内取定一个值,使得各变量在当前的取值情况下满足所有给定的约束关系.一个 CSP 的所有解构成该问题的解空间.在实际应用中,经常要求各变量的值域有限,这时,该类 CSP 称作约束满足网(constraint satisfaction network,简称 CSN),也称为约束网络.对约束网络的研究,有时限定在给定的约束关系族中,每一个约束关系至多涉及两个变量,此时,这类 CSN 称作二项约束网络.

CSP 在组合优化、硬件设计、人工智能等方面有着广泛的应用.一般情况下,CSP 是一类 NP-完全问题[1].但是,如果对 CSP 的约束关系加以必要的限制,CSP 是多项式可解的[2~4].在实践中,对 CSP 主要有两类研究方向:利用一致性技术寻找特定 CSP 的一个解或解空间;采用无回溯算法求解 CSP,将约束网络最小化.若利用一致性技术将不可能是 CSP 可行解的各变量的取值从约束网络中清除.

对二项约束网络,文献[5]提出了一类易于求解的二项约束网络:RC 约束网络,同时给出了其相应的快速识别算法.文献[5,6]构造了与 RC 约束网络相关但又有区别的二项约束网络——CRC 约束网络,但没有给出 CRC 约束网络的识别算法.本文在文献[5,6]的基础上,对 CRC 约束网络的识别进行了进一步的研究.

1 相关定义

定义 **1**. 二项约束网络.

* 收稿日期: 2000-06-10; 修改日期: 2001-03-01

基金项目: 国家自然科学基金资助项目(60005004);国家重点基础研究发展规划 973 资助项目(G1998030509)

作者简介: 陈恩红(1968－),男,安徽宁国人,博士,副教授,主要研究领域为约束满足问题,知识发现,机器学习;张振亚(1972－),男,安徽淮北人,博士生,主要研究领域为约束满足问题,知识发现;王煦法(1948－),男,江苏丹阳人,教授,博士生导师,主要研究领域为智能信息处理.

二项约束网络可以定义为 $N=(Var,D,C)$,其中

(1) $Var=\{x_1,x_2,\ldots,x_n\}$,$x_i$ 称作变量,$i=1,2,\ldots,n$;

(2) $D=\bigcup_{i=1}^{n} D_i$,$x_i\in D_i,i=1,2,\ldots,n$,$D_i$ 称为变量 x_i 的定义域;

(3) $C=\{C_{ij}|x_i,x_j\in Var,C_{ij}\in D_i\times D_j,i,j=1,2,\ldots,n\}$.

在应用中,通常把 $(v,w)\in C_{ij}$ 记作 $C_{ij}(v,w)$.对给定的二项约束网络 $N=(Var,D,C)$,有以下简记:$d=\max\{|D_i||i=1,2,\ldots,n\}$,$arc(N)=\{(i,j)|C_{ij}\in C,i,j=1,2,\ldots,n\}$.

对 $N=(Var,D,C)$,通常假定 D_i 是有序集,有时甚至假定 D 是有序集.对二项约束网络 N,由于约束关系 C_{ij} 满足 $C_{ij}\in D_i\times D_j$,故 C_{ij} 可用 $C_{ij}\in D_i\times D_j$ 上的关系矩阵 M_{ij} 表示.$\forall\, v\in D_i,w\in D_j,M_{vw}=1$ 当且仅当 $C_{ij}(v,w)$,否则,$M_{vw}=0$.在无特别说明的情况下,认为 C_{ij} 本身便是关系的表示矩阵.

定义 **2**. 二项约束网络的解.

设 $N=(Var,D,C)$,则$(v_1,v_2,\ldots,v_n)$是 N 的一个解当且仅当对$\forall\,(i,j)\in arc(N),(v_i,v_j)\in C_{ij}$.

定义 **3**. 行凸性(RC).

设 $N=(Var,D,C),C_{ij}\in C,C_{ij}$ 是关系矩阵.若在 C_{ij} 的每一行中"1"的出现是连续的,即在任一行中,取值为"1"的列中间无取值为"0"的列,则 C_{ij} 是行凸的.

定义 **4**. 空行/列.

设 M 是一布尔矩阵,称 M 的取值全为 0 的列(行)为 M 的空列(行).

定义 **5**. C_{ij}^*.

在 $N=(Var,D,C)$中,设 C_{ij} 是关于变量 x_i,x_j 的约束关系的矩阵表示,C_{ij}^*是 C_{ij} 的简化或精简形式.其中 C_{ij}^*由将 C_{ij} 中的空行、空列移去后得到.在约束关系 C_{ij} 中,变量 x_i 的可以取值的值域记作 $D_i(C_{ij})$,即 $D_i(C_{ij})=\{v|\exists\, w\in D, C_{ij}(v,w)\}$.

定义 **6**. 映像、前驱、后继.

设 $N=(Var,D,C),C_{ij}\in C$ 且 C_{ij} 行凸,$v\in D_i(C_{ij})$,则 $Image(v)=\{w|<v,w>\in C_{ij}\}$称作 v 的映像.

由于 C_{ij} 行凸,$Image(v)$可以用 $D_j(C_{ji})$上的区间$[w_1,w_m]$表示.其中

$$w_1=\min(\{w|C_{ij}(v,w)\});$$
$$w_m=\max(\{w|C_{ij}(v,w)\}).$$

在给定的 $C_{ij},\forall\, w\in D_j(C_{ji}),Pred(w,D_j(C_{ji}))$表示 w 在 $D_j(C_{ji})$的前驱,$Succ(w,D_j(C_{ji}))$表示 w 在 $D_j(C_{ji})$的后继.在 $D_j(C_{ji})$确定的情况下,$D_j(C_{ji})$可以省略,即以 $Succ(w),Pred(w)$表示 w 在当前的后继和前驱.

定义 **7**. 连接性.

设 $N=(Var,D,C),C_{ij}\in C$ 且 C_{ij} 是关系的矩阵表示且行凸,C_{ij}^*的任一连续的两行的映像为$[a,b]$和$[a',b']$.若 $b'\geq Pred(a)\wedge a'\leq Succ(b)$,则 C_{ij} 是相接的.

显然,任意两行非连接的情况为 $\neg(b'\geq Pred(a)\wedge a'\leq Succ(b))$,即 $b'<Pred(a)\vee a'>Succ(b)$.故对任意连续的两行,若其映像为$[a,b]$和$[a',b']$,$[a,b]\cap[a',b']=\varnothing$,则两行必不连接.

定义 **8**. 相接行凸(CRC).

设 $N=(Var,D,C),C_{ij}\in C.C_{ij}$ 是相接行凸的当且仅当 C_{ij}^*,C_{ji}^*是行凸的;C_{ij}^*,C_{ji}^*是相接的.

文献[5]已证明:若 C_{ij}^*,C_{ji}^*是行凸的,C_{ij}^*是相接的当且仅当 C_{ji}^*是相接的.故在验证关系矩阵的相接性时只需对 C_{ij}^*,C_{ji}^*中的一个进行.概念属性特征及其分类.

从 RC 和 CRC 的定义知:由于在 CRC 中可能存在空列(行),CRC 约束不一定是 RC 约束,但如果将 CRC 约束进行简化,则简化后 CRC 的必然是 RC 的;另一方面,由于 RC 约束不必满足连接性质,故 RC 约束不一定是 CRC 的.图 1 可以直观显示 RC 与 CRC 的区别.图 1 为二项约束的矩阵表示,其中阴影所在的元素表示 1,空白表示 0.在图 1 中的 RC 约束不是 CRC 约束,CRC 约束不是 RC 约束.

RC constraints①　　CRC constraints

①约束.

Fig.1 Matrix representation of binary RC and CRC constraints

图 1 RC 与 CRC 的二项约束的矩阵表示

文献[6]在判定 RC 约束网络时,为降低算法的时间复杂度采用了基于 PQ-树的算法[7].对给定集合 $U=\{u_1, u_2,\ldots,u_m\}$.所谓 PQ 树是指建立在 U 上的一棵有序树,其中,树的叶子节点由 U 的所有元素构成,一片叶子对应且仅对应 U 中的一个元素;树的内部节点有两类:P 节点和 Q 节点.对 PQ 树中只能进行基于模式匹配的剪枝(reduce)操作.实践中,通常将剪枝操作分成两步:Bubble 和 Reduce.Bubble(T,S)用来根据集合 S 来标记 PQ 树 T 中需要进行剪枝操作的节点,Reduce(T,S)根据 Bubble(T,S)的标记来完成剪枝操作.

利用 PQ 树判定一个矩阵 M 是否可以经过有限的列变换使得每一行中的 1 连续出现(算法 1).对于算法 1,如果 M 是 $m\times n$ 阶(0,1)-矩阵,M 中非 0 元素的个数为 f,则时间复杂度为 $O(m+n+f)$.

算法 **1**. 判定可否使 M 的每一行中的 1 连续出现.

```
Boolean Procedure Consecutive(M)
 Begin
  Let U be the set of column in M;
  T=T(U,U);
  For j=1 to n      //n 是 M 的行数
     Begin
        Let S be the set of columns which have a one in row j;
        T:= Bubble(T,S);
        T:=Reduce(T,S);
        if T=T(φ,φ) then return false
     End
  Return true
 End
```

对 $N=(Var,D,C)$,文献[6]给出了是否存在变量以及变量值域中元素的重新排列使得 $N=(Var,D,C)$是行凸的的算法 Findorder(C,n).其时间复杂度为 $O(n^3d^2)$,其中,n 为约束关系涉及的变量数,即 $n=|Var|$,d 为各变量的定义域中最大定义域的大小,即 $d=\max\{|D_i|,i=1,2,\ldots,n\}$.

2 矩阵的索引表示和 CRC 标准型

2.1 矩阵的索引表示

由于需要对矩阵进行频繁的行列互换操作,为减少因之操作的时间,可以将矩阵用索引表示法表示,这样,在对矩阵的行列互换时,不必交换矩阵的实际行列的元素,只要修改其索引即可.为索引表示矩阵,需要为矩阵建立两个索引表,在实际编程时,可以将这两张表封装在存储矩阵的数据结构中.

例 1:矩阵的索引表示.

若 $A=\begin{bmatrix}1111\\1110\\1100\\1000\end{bmatrix}$,$A'=\begin{bmatrix}1111\\1011\\1001\\1000\end{bmatrix}$,$A'$是由 A 的 2、4 列交换后得到,则

RowIndex(A)=<(1,1),((2,2),(3,3),(4,4)>,ColumnIndex(A)=<(1,1),(2,2),(3,3),(4,4)>,

RowIndex(A)=<(1,1),((2,2),(3,3),(4,4)>,ColumnIndex(A)=<(1,1),(2,4),(3,3),(4,2)>,

其中 RowIndex(A)为 A 的行索引表,其元素形如(a,b),a 表示当前矩阵的行号,b 表示当前矩阵相应行对应的初始矩阵的相应行的行号.ColumnIndex(A)为 A 的列索引表,其元素形如(a,b),a 表示当前矩阵的列号,b 表示当前矩阵相应列对应的初始矩阵的相应列的列号.

算法 **2**. 索引化矩阵的行、列交换.

```
void Swap_Matrix (M,flag,n1,n2)    //M 是矩阵,flag 标记行或列,n1,n2 是将交换的行或列
Begin
   int temp;
   list *worklist;        //根据 flag 的值指向行索引或列索引
   if (flag==ROW) worklist = M.RowIndex;
                  else   worklist = M.ColumnIndex;
   temp = worklist[n1].b;
   worklist[n1].b = worklist[n2].b;
   worklist[n2].b = temp
End
```

算法 2 是索引化矩阵的行列交换的算法.显然,若欲进行行列操作的矩阵每行或每列有 n 个元素,则算法 2 的时间复杂度为 $O(1)$,而交换矩阵实际元素的算法的时间复杂度至少为 $O(n)$.矩阵索引化后,上述算法即可使用,之前需要对矩阵进行索引化.设被索引的矩阵的阶为 $m\times n$,则对其索引化时只需要进行 $m+n$ 个对索引节点的写操作.由于对任一矩阵只需索引化一次,故因矩阵的行列交换所使用的时间将显著降低.

算法 **3**. 矩阵的索引化算法.

```
void indexmatrix(M)    //M 是矩阵
Begin
   M.rowindex = (list*)mcalloc(indexnode,M.rowsize);
   M.columnindex = (list*)mcalloc(indexnode,M.columnsize);
   For(i=0;i<M.rowsize;i++)    M.rowindex[i].a = M.rowindex[i].b = i;
   For(i=0;i<M.columnsize;i++)    M.rowindex[i].a = M.columnindex[i].b = i
End
```

2.2 CRC约束矩阵的标准型

设 M 是 $m\times n$ 阶(0,1)矩阵,CRC 性质满足.M 的第 i 行用 i 表示,则第 i 行的映像为 $Image(i)$.min($Image(i)$)表示第 i 行中第 1 个非零元素所在的列的列号,max($Image(i)$)表示第 i 行中最后一个非零元素所在的列的列号.

若满足:

性质 1. $\min(Image(1))\le\min(Image(2))\le\ldots\le\min(Image(m))$;

性质 2. $\max(Image(i))\le\max(Image(i+1))\le\ldots\le\max(Image(i+k))$,

若 $\min(Image(i))=\min(Image(i+1))=\ldots=\min(Image(i+k))$,其中 $1\le i,i+k\le m$;

则称 M 是标准的 CRC 矩阵.具有性质 1,性质 2 的 CRC 矩阵称作 CRC 矩阵的标准型.

对任一 CRC 矩阵 C 可以进行标准化操作,操作的结果矩阵 C'是 C 的行经过互换得到.在互换 C 的行时,标准型的性质 1,性质 2 应遵守,同时要求如果对 CRC 矩阵 C 有 $\max(Image(i))=\max(Image(i+1))=\ldots=$

$\max(Image(i+k))$且 $\min(Image(i))=\min(Image(i+1))=\ldots=\min(Image(i+k)),1\le i,i+k\le m$ 则在 C'中第 $i,i+1,...,i+k$ 行的顺序保持原顺序.

命题 **1**. 任意一个 CRC 矩阵标准化后仍是 CRC 的.

证明:设 C 是 CRC 矩阵,row(C)表示 C 的行数,C'表示 C 对应的标准型.

以下对 row (C)归纳

(1) row(C)=1 时,$C=C'$,显然成立;

row(C)=2 时,若 C 已经是标准的,则 $C=C'$,显然成立,否则,标准化时需将两行对换,对换不影响两行间的连接性关系,由 C 是 CRC 的得知,C'也是 CRC 的.

(2) 设 row(C)=k 时,命题成立,则当 row(C)=k+1 时,设[a,b]是区间,Turbo([a,b])定义为[pred(a),succ(b)];

① 若 $\min(Image(k+1))=\min(\{\min(Image(i))|i=1,2,\ldots,k\}$则

设 C_k是 C 的前 k 行构成的子阵,由 C 是 CRC 的得知,C_k是 CRC 的.

由于 row(C_k) =k,故可以将 C_k 标准化为 C_k',且 C_k'是 CRC 的.

设 C 的第 $k,k+1$ 行的映像分别为 $Image(k)$和 $Image(k+1)$,C_k'的第 k 行的映像为 $Image(k')$,由标准化操作可知,$Image(k)\le Image(k')$;

又由连接性的定义可知:

$$\mathrm{Turbo}(Image(k))\cap Image(k+1)\neq\varnothing,$$

故

$$\mathrm{Turbo}(Image(k'))\cap Image(k+1)\neq\varnothing,$$

所以将 C 的第 $k+1$ 行作为 C_k'的第 $k+1$ 行加入 C_k'后,C_k'仍然是 CRC 的.由 C_k'的构造过程得知,C_k'后即为 C'.

② 若 $\min(Image(k+1))\neq\min(\{\min(Image(i))|i=1,2,\ldots,k\}$,则:

设 C_k是 C 的前 k 行构成的子阵,由 C 是 CRC 的得知,C_k是 CRC 的.

由于 row(C_k)=k,故可以将 C_k 标准化为 C_k'且 C_k'是 CRC 的.设 C_k'的每一行的行号为 $1',2',\ldots,k'$,由标准型的定义知 $\min(Image(1'))\le\min(Image(2'))\le\ldots\le\min(Image(k'))$.

设 C 的前 k 行的映像分别为 $Image(1),...,Image(k)$,则

$$\{\min(Image(1')),\ldots,\min(Image(k'))\}=\{\min(Image(1')),\ldots,\min(Image(k'))\}$$

又∵

$$\min(Image(k+1))\neq\min(\{\min(Image(i))|i=1,2,\ldots,k\}$$

∴

$$\min(Image(k+1))\neq\min(\{\min(Image(i'))|i=1,2,\ldots,k\}$$

∴

$$\exists i\in\{1,2,...,k-1\},\min(Image(i'))\le\min(Image(k+1))\le\min(Image((i+1)'))$$

又∵ C_k'是 CRC 的,

∴

$$\mathrm{Turbo}(Image(i'))\cap Image((i+1)')\neq\varnothing$$

∵

$$\min(Image(i'))\le\min(Image(k+1))$$

∴

$$\mathrm{turbo}(Image(k+1))\cap Image((i+1)')\neq\varnothing$$

∴将 C 的第 $k+1$ 行插入 C_k'的第 i'和$(i+1)'$之间后 C_k'仍然是 CRC 的.

又由 C_k'的构造知其为 C 的标准型.

综合①,②得知,Row(C)=k+1 时命题成立

综合(1),(2)得知,对任意的自然数 m,Row(C) = m 时命题成立.

命题 **2**. CRC 标准化在有限步内可以完成.

证明:将 CRC 矩阵 C 各行按标准化操作要求重新排序即可.

命题 **3**. 任一 RC 矩阵,若标准化后是 CRC 矩阵,则该矩阵经过一系列的列变换后必然是 CRC 矩阵.

证明:CRC 标准型是 CRC 的;CRC 标准化在有限步内可以完成.

命题 **4**. 任何 CRC 矩阵经过有限步列变换可以转化为 RC 矩阵.

证明:由 CRC 的定义知,CRC 与 RC 的唯一区别在于其精简形式中,CRC 在 RC 定义的基础上附加了两行间元素取值的连接性的约束.故在精简形式下,CRC 是 RC 的.而 CRC 的简化形式可以由矩阵经过有限步行列变换后去除空行、列得到,其中的列变换操作使 RC 性质满足,行变换使连接性满足.所以,任何 CRC 矩阵经过有限步

列变换可以转化为 RC 矩阵.

2.3 CRC的识别

为了识别 CRC 约束矩阵,需要将算法 1 的返回值由布尔型改为 PQ 树,修改后的算法 4 如下:

算法 **4**. 判定 M 在 T 允许的列变换下,可否使其每一行中的 1 连续出现.

```
//T 是 PQ 树
PQTree  Procedure Consecutive(M,T)
 Begin
   For j=1 to n    //n 是 M 的行数
     Begin
       Let S be the set of columns which have a one in row j;
       T:=Bubble(T,S);
       T:=Reduce(T,S);
       If T=T(∅,∅) then return NULL
     End
   Return T
 End
```

根据算法 Findorder 与算法 4 以及对 CRC 性质的进一步讨论,有如下关于“对 $N=(Var,D,C)$,是否存在 Var 中的变量的某种排列以及各变量值域组成元素的重新排列使得 C 中的元素在重排后均有 CRC 特性”的算法.

算法 **5**. CRC 约束网络的判定.

```
      CRC_Findorder (C,n)
①    L={1,2,…,n}       //n 为变量数
②    For m = n downto 1
③    Do
④      寻找一个 j∈L,使得 D_j 中的元素存在排列满足: ∀ i∈L,C_ij* 是行凸的(若这样的 j 不存在,报错返回);
        //在用算法 4 验证 C_ij 是行凸性时,参数 T 为本轮查找上一次算法 4 的返回值,
        //第 1 次调用算法 4 时,T = T(D_j,D_j)
⑤      For i=1 to n do
⑥        If C_ij∈C then 将 C_ij* RC 化;
⑦                      将 RC 化后的 C_ij* 标准化;
⑧                      验证标准化后的 C_ij* 连接性是否成立,不成立则报错返回;
⑨      将变量 X_j 的位序标记为 m;
⑩      L=L−{j}
      End
```

命题 **5**. 给定一约束网络 C,算法 5 是正确的,即该算法可以判定 C 是否是 CRC 的.

证明:充分性.

不考虑算法 5 的第⑤~⑧句,算法可以判定对 RC 约束网络 $N=(Var,D,C)$,是否存在 Var 中的变量的某种排列以及各变量值域组成元素的重新排列使得 C 中的元素在重排后均有 RC 特性(参见文献[6]中相应的证明).第⑤~⑧句是对 RC 化后的约束矩阵的精简形式的标准型连接特性进行判定.由命题 3 若连接特性成立则相应的 RC 矩阵可以 CRC 化.

必要性.

从 RC 和 CRC 的定义可知,由于在 CRC 中可能存在空列(行),CRC 约束不一定是 RC 约束,但如果将一 CRC 约束进行简化,由命题 4 可知简化后的 CRC 必然是 RC 的.

命题 **6**. 算法 5 的时间复杂度为 $O(n^3d^2)$,其中 $n=|Var|,d=\max\{|D_i||i=1,2,\ldots,n\}$.

证明:不考虑算法的第⑤~⑧句,算法 5 的时间复杂度为 $O(n^3d^2)$[6].第⑤~⑧句是对 C 中的每一个约束关系进行考虑.因为 C 中的每一个约束关系都是二项约束关系,故$|C|\le\binom{n}{2}$,即$|C|=O(n^2)$,对每一个约束矩阵,第⑥,⑦句的时间复杂度在用索引法表示矩阵时为 $O(d)$,第⑧句可以在 $O(d^2)$时间内完成.所以,算法 5 的时间复杂度为

$$O(n^3d^2)+O(n^2)\times[O(d)+O(d)+O(d^2)]=O(n^3d^2).$$

文献[5]在研究 CRC 约束矩阵时虽然没有给出 CRC 约束网络的识别算法,但是根据文献[6]的相关工作明确指出了识别算法时间复杂度应为 $O(n^3d^2)$并以其作为可以进一步研究的问题,而算法 5 明确肯定了这一结论.

3 结 论

本文通过定义CRC约束矩阵的标准型及对其性质的分析,在矩阵的索引表示法下,给出了CRC约束网络的快速识别算法.该算法达到了文献[5]指出的此类算法最优时间复杂度.我们将对有关 CRC 约束矩阵标准型的代数性质,如通过约束矩阵和标准型的数字特征的计算对 CRC 约束网络的识别作进一步的研究.此外,对 CRC 约束网络可以使用无回溯搜索算法求解相应的解空间或解,将现实中的 CSP 问题 CRC 约束网络化应该是很有意义的实践.

References:

[1] Mackworth, A.K. Consistency in networks of relations. Artificial Intelligence, 1977,8(1):99~118.

[2] Dechter, R., Pearal, J. Network-Based heuristics for constraint satisfaction problem. Artificial Intelligence, 1988,34(1):1~38.

[3] Montanari, U. Networks of constraints: fundamental properties and application to picture processing. Information Science, 1974, 7(1):95~132.

[4] Henterryck, P.V., Deville, Y., Teng, C.M. A generic arc-consistency algorithm and it's specializations. Artificial Intelligence, 1992, 57(2):291~321.

[5] Beek, P.V. On the minimality and global consistency of row-convex constrain networks. Journal of the Association for Computing Machinery, 1995,42(3):543~561.

[6] Deville,Y., Barette, O., Henterryck, P.V. Constraint satisfaction over connected row-convex constraints. Artificial Intelligence, 1999,109(2):243~271.

[7] Booth, K.S., Lucker, G.S. Testing for the consecutive ones property, interval graphs and graph planarity using PQ-Tree algorithm. Journal of Computer and System Sciences, 1976,13(2):335~379.

A Fast Recognition Algorithm of CRC Constraint Networks*

CHEN En-hong, ZHANG Zhen-ya, WANG Xu-fa

(*Department of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027, *China*)

E-mail: cheh@ustc.edu.cn

http://mail.ustc.edu.cn/~zzychm

Abstract: Constraint networks provide a useful framework for the formulation of many problems in computer science. In general, constraint satisfaction problems are NP-Complete. Many problems specially restrict the structure of constraints or the form of the constraints, then allow them to be solved efficiently. For the identification of such tractable constraints, much work has been devoted to the study of special classes of constraints or constraint networks. As pointed out by Deville *et al.*, a class of connected row-convex (CRC) constraints is shown to be tractable. This paper intends to find an efficient recognition algorithm for the class of constraint networks. In this paper, a standardized form for the CRC constraint matrix is proposed based on the related findings on CRC constraint networks. The basic characteristics of the standardized form are analyzed, and a fast algorithm is provided for the recognition of CRC constraint networks based on a recognition algorithm of the RC constraint network, properties of PQ tree (a tree composed by P nodes and Q nodes)and an indexed matrix representation of constraints. The time complexity of the algorithm is $O(n^3d^2)$, where n is the number of variables in a constraint network and d is the maximum size of the domain for each variable. It reaches the optimum time complexity for a CRC constraint network recognition algorithm, and hence provides a feasible solution to practical CRC constraint satisfaction problems.

Key words: binary constraint network; RC constraint network; CRC constraint network PQ-Tree; the standardized form for CRC constraint matrix

---

* Received June 10, 2000; accepted March 1, 2001

Supported by the National Natural Science Foundation of China under Grant No.60005004; the National Grand Fundamental Research 973 Program of China under Grant No.G1998030509