

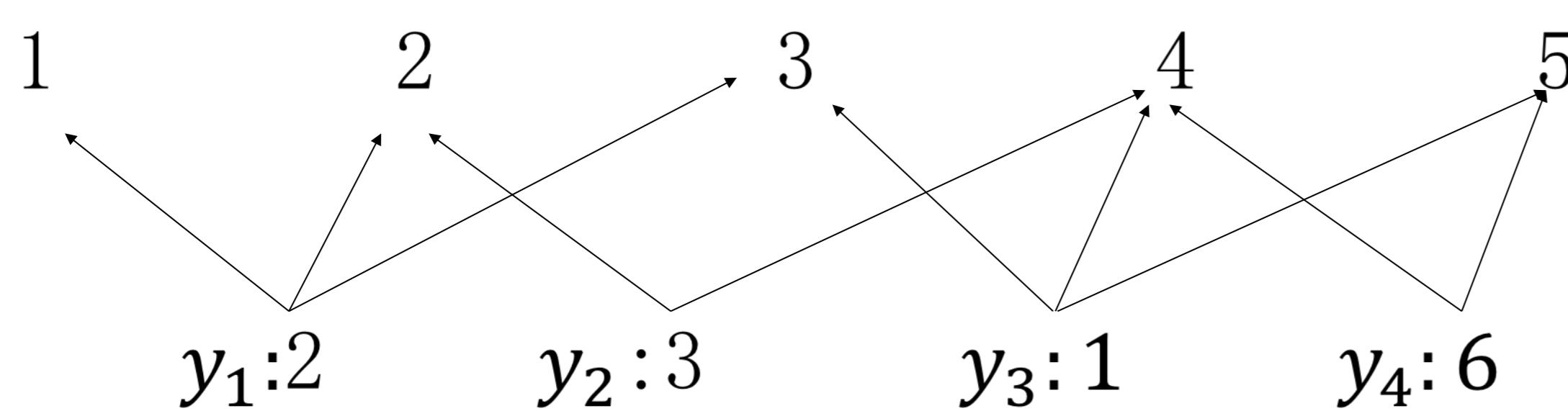
MaxSAT求解集合覆盖与顶点支配集问题

雷震东，蔡少伟

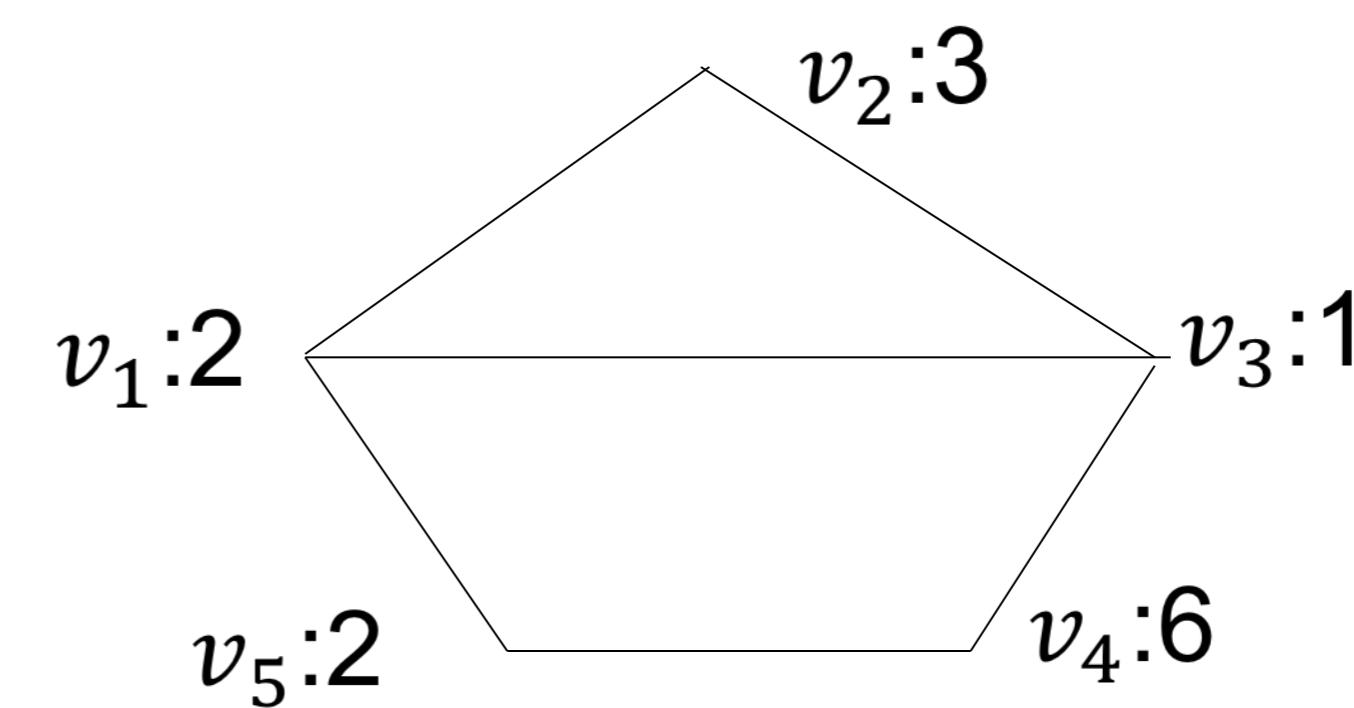
Solving Set Cover and Dominating Set via Partial MaxSAT.

AAAI2020, zhendong lei, leizd@ios.ac.cn, 15600038788

集合覆盖: 给定集合 $X=\{1, 2, \dots, n\}$, 以及集合 $Y=\{y_1, y_2, \dots, y_m\}$, 每个 y_i 是 X 的某个子集, 其权值为 $w(y_i)$ 。目标是找到一个 Y 的子集 F , 使得 F 覆盖了 X 中的所有元素, 并且使得 F 中的集合权值之和最小。例如, 左图中 $F=\{y_1, y_3\}$ 是最小集合覆盖, 其权值和为: $2+1=3$



顶点支配: 给定无向图 $G=(V, E, W)$, 每个顶点有权值 $w(v)$, 目标是找到一个子集 $D \subseteq V$, 满足任意的点 $v \in V$, 要么 v 本身属于 D , 要么 v 的一个邻居属于 D 。并且使得 D 的顶点权重之和最小。如下图: $D=\{v_1, v_3\}$ 是最小顶点支配, 其权值和为: $2+1=3$



研究意义:

- 集合覆盖和顶点支配问题是经典的NP-hard问题, 在卫星定位, 铁路路网规划等领域有重要作用。现实的工业应用中, 问题规模十分庞大, 解决这类问题十分困难。
- 相比于其他数学模型, 编码成MaxSAT十分自然紧凑, 且得到的实例具有特殊的性质。另外, MaxSAT是一种可推理的逻辑语言, 根据这些独特的性质可以设计专门的推理规则来消减问题规模。

主要贡献:

- 将集合覆盖与顶点支配问题编码成MaxSAT, 点出其实例特性, 设计推理消减规则以及高效的局部搜索算法。
- 实验结果表明, 消减规则极大的消减了实例规模, 算法性能优于所有传统的集合覆盖, 定点支配以及MaxSAT算法。

Partial MaxSAT问题:

- Partial MaxSAT (PMS) 是MaxSAT问题的一般化, 其子句分为硬子句和软子句, 每个软子句有其对应权重。目标是满足所有的硬子句, 并且最大化满足的软子句权重之和。
- 给定布尔变量集 $\{v_1, v_2, \dots, v_n\}$, 文字是一个布尔变量 v_i 本身或者其否定 $\neg v_i$, 每个子句可以看作是文字的集合, 一个子句被满足当且仅当至少有一个其包含的文字为真。由集合覆盖和顶点支配集问题编码的PMS实例有以下性质:
- 1. 所有硬子句只包含正文字。2. 对于每个变量 v_i , 都一一对应一个软子句 $(\{\neg v_i\}, w_i)$, 其中 w_i 为软子句 $\{\neg v_i\}$ 的权值。

推理规则:

- 支配关系: 如果所有包含 v_1 的硬子句都包含 v_2 , 则 v_2 支配 v_1 。若此时 v_1, v_2 对应软子句 $\{\neg v_1\}, \{\neg v_2\}$ 权重满足 $w_1 \geq w_2$, 则 v_2 强支配 v_1 。
- 单元硬子句规则: 如果存在单元硬子句 $\{v\}$, 则 v 被赋值为1, 并且将有包含 v 的硬子句(已被满足)删除。
- 强支配规则: 如果变量 v_2 强支配 v_1 , 则 v_1 被赋值为0。删除软子句 $\{\neg v_1\}$, 移除所有硬子句中的文字 v_1 。
- 二元弱支配规则: 如果存在一个二元硬子句 $\{v_1, v_2\}$, 且 v_2 支配 v_1 , 删除所有包含 v_1 的硬子句, 删除软子句 $\{\neg v_1\}, \{\neg v_2\}$ 的权重改为 $w_2 = w_2 - w_1$ 。

一个PMS实例:

硬子句:	软子句:
$h1:\{v_1, v_2\}$	$\{\neg v_1\}, 2$
$h2:\{v_1, v_2, v_3, v_4\}$	$\{\neg v_2\}, 3$
$h3:\{v_2, v_3, v_5\}$	$\{\neg v_3\}, 2$
$h4:\{v_2, v_4, v_5\}$	$\{\neg v_4\}, 2$
$h5:\{v_3, v_4, v_5\}$	$\{\neg v_5\}, 7$
$h6:\{v_5, v_6, v_7\}$	$\{\neg v_6\}, 5$
$h7:\{v_6, v_7, v_8\}$	$\{\neg v_7\}, 3$
$h8:\{v_7, v_8\}$	$\{\neg v_8\}, 4$

根据强支配规则: v_8 被赋值为0, 删除 $\{\neg v_8\}$, $h7$ 和 $h8$ 分别变为 $\{v_6, v_7\}, \{v_7\}$ 。
 根据单元硬子句规则: v_7 被赋值为1, 删除 $h6, h7, h8$, 软子句 $\{\neg v_7\}$ 变为空子句。空子句永远是不满足的。
 根据弱支配规则: 删除 $h1, h2, h3, h4, h5, h6$, $\{\neg v_1\}$ 变为空子句, $\{\neg v_2\}$ 的权重由3变为1。

局部搜索算法: 通过贪心算法找一初始解, 然后迭代改进, 每一步反转一个变量的值。主要技术如下:

1. 采用hard score和soft score分别评估操作对满足硬约束和软约束的数量变化, 并据此设计了四种搜索模式, 包括下降模式, 探索模式, 渴求模式和扰动模式。
2. 设计“性价比”函数作为变量的主要选择标准:

$$\text{score}(v) = \text{hard score}(v) / |\text{soft score}(v)|$$
3. 通过MaxSAT建立变量关联关系。

实验结果:

- 推理规则可平均消减60%以上的变量和子句。
- 我们的算法几乎在所有的大规模工业实例上的性能都要优于所有的顶点支配, 集合覆盖以及MaxSAT算法。具体实验结果见下表:

