AAS-CN-2010-0446 稿件“CP_nets 的表达能力”的修改说明

首先感谢责任编辑和审稿专家对该篇论文所提的意见和建议。为了使稿件更加完善和精炼，作者结合专家的意见和建议对原稿进行了修改，请各位专家批评指正。下面是对审稿专家意见的一些处理说明。

关于第 1 位审稿专家的意见：

一、关于审稿专家的“若题目改为“CP_nets 及其表达能力研究”可能更为合适，也已突出文章在表达能力分析方面的独到性，且不失其综述的特点”的处理

处理方法：将题目修改为“CP_nets 及其表达能力研究”。

原因：虽然原稿以 CP-nets 的表达能力为线索在各个章节中来阐述 CP_nets 结构的基本数学性质与应用，但是第 2，3 和 5 大节从标题上看都仅是在论述 CP_nets，因此题目更改后，读者更容易把握论文阐述的观点及其重点。

二、关于审稿专家的“对于文章所声明的重点研究问题：CP_nets 的表达能力（第四节）着墨有些单薄，论证相对简略，也缺乏较为直观的图表和实例。”的处理。

限于篇幅，本文的第 4 节着墨有些单薄，但结合专家的建议，在修改后的论文中增加了一些论述，具体有：

1. 增加了性质 1 的证明。

原因：原论文中的性质 1 仅仅做了阐述，而未加证明。虽然该性质相对比较简单，但由于性质中自然语言描述的成分较多，特别是“直观”的含义，且数学描述成分相对少，因此容易产生歧义。为此在修改后的论文中增加了性质 1 的证明，其目的不仅在于证明，关键在于进一步刻画性质描述的内容。

2. 增加了定理 10 描述的相关论述和参考文献[24]。

原因：定理 10 描述 CP-nets 可处理多属性的定性偏好决策，修改后的论文在定理 10 后增加了对定量偏好和定性偏好的阐述，并增加了参考文献[24]来说明定性偏好与定量偏好大量的应用及特点的出处。

关于第 2 位审稿专家的意见：

一、关于审稿专家的“1.第 2.4 节，第一段，“利用它 Agent 不用反省就可得到其含义”这里需要说明“反省”准确含义”的处理

处理方法：在修改后的论文增加了“反省”的含义。反省就是仔细思考的意思。 CP_nets的主要特点之一就是其遵循ceteris paribus的语义，其英语含义为“all other things being equal”，其中文含义为“其他属性值都一样的话”。Agent之所以不用反省就在于对于两个可交换的配置o_1与o_2，由于它们只有一个属性值不同，而其余属性值都一样，因此o_1与o_2孰优孰劣的判断agent看一眼便知，不用仔细思考和推理。

事实上论文 2.4 节的第二段在解释 ceteris paribus 语义的同时，也在解释不用反省的含义，也在解释 CP_nets 的特点，即对于可交换的两个配置，Agent 不用思考就能得到孰优孰劣的判断。

原因：原论文在个别概念和术语的描述上口语化较重，虽然含义明确，但具体到 CP_nets 的应用上，含义就不一定清楚，为此增加“反省”的含义，并结合 CP-nets 的语义进行了解释。

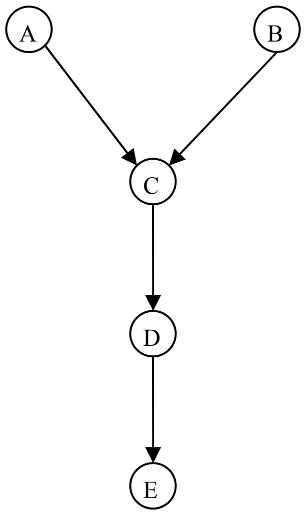
二、关于审稿专家的“ 2. 例 2 仅对 Ceteris Paribus 语义做了解释，应放在定义 7 之前。”的处理

处理方法： 将例 2 放到了定义 7 之前。

原因：因为例 2 仅对 Ceteris Paribus 语义做了解释。原论文想在例 2 中不仅对 Ceteris Paribus 语义作解释，还想对定义 7 作解释。但为了不增加篇幅，就放弃了对定义 7 的解释，导致例 2 放的位置不太合适。

三、关于审稿专家的“3.需要一个例子来解释定义 7.”的处理

处理方法：在修改后的论文增加了新的实例“例 3”。为了节省篇幅，例 3 在修改后论文中只给出了其代数表示方法，没有给出类似于图 2 的图形表示方法。修改后的论文中例 3 的 CP_nets 的图形表示法如下：



修改后的论文中例 3 的 CP_nets 的图形表示

事实上定义 7 所阐述的条件偏好无关概念有许多重要性质（例如两个属性集是条件偏好无关的，则其在 CP-nets 中的结构有一些特点等）。限于篇幅，本文只给出其定义，以帮助理解 CP_nets 的 cateris Paritus 语义。

原因：定义 7 给出了条件偏好无关的概念，它是 CP_nets 的 Ceteris Paribus 语义的具体体现，具体是通过 CP_nets 中的条件偏好表来体现的。其作用在于为 5.3.3 节给出的规约的特性作准备。

四、关于审稿专家的“ 4.定义 9 中出现的强占优 “N |=o > o’”，是在定义 10 中定义的，故定义 10 应在定义 9 之前。”的处理

处理方法： 把原论文中的定义 10 放到了定义 9 之前，即将定义 9 和定义 10 互换了。

原因：遵循科技论文中“先定义，后引用”的原则。

五、关于审稿专家的“ 5.图 4 描述的算法中内层循环应该有更多的缩进。”的处理

处理方法：在修改后的论文中，对图 4 描述的算法的内层循环增加了更多的缩进。

原因：能清晰地理解程序中外层循环和内层循环的作用区域，从而能正确理解算法的操作步骤。

六、关于审稿专家的“6.定理 4 中 “使得 o 和 o’ 的属性 X 的祖先值相同” 指的是 X 的所有祖先还是 X 一个祖先？ ”的处理

处理方法：定理 4 中“*o* 和 *o*’ 的属性 X 的祖先值相同”指的是 X 的所有祖先。

原因：因为定理 4 的证明核心是：找 *o* 和 *o*’ 的首个不相同的属性值，若该属性值对应的属性为 X，则在此之前的所有的属性值（排在拓扑排序序列中属性 X 之前的所有属性值）都是相同的，因此 *o* 和 *o*’ 的属性 X 的所有祖先值都相同。

七、关于审稿专家的“ 7.看不出定理 4 是如何证明的。”的处理

处理方法： 在修改后的论文中，在定理 4 的脚注中增加了证明思路，并在证明步骤中增加了部分操作的解释。

其实将两个配置 *o* 和 *o*’看成长度相同的两个字符串，则定理 4 的核心是找 *o* 和 *o*’中首个不相同的字符，若在首个不相同的字符处，*o* 对应的属性值优于 *o*’对应的属性值，则 *o*’不会比 *o* 强，即 $N \nvDash o' > o$；若 *o*’对应的属性值优于 *o* 对应的属性值，则 *o* 不会比 *o*’强，即 $N \nvDash o > o'$。

原因：若一个定理的证明思路清楚后，证明过程就只是思路的描写过程，也容易看出定理的实质和证明的思想了。

八、关于审稿专家的“8.需要一个例子来解释定理 4。”的处理

处理方法：在修改后的论文增加了实例“例 5”。

原因：以例子来理解定理是计算机科学的一个基本思路，从例子可清楚看出定理所要阐述的内容和证明的过程。

九、关于审稿专家的“9.定理 9 的证明中，没有从理论上说明“某个 CP_nets 导出图求了传递闭包后元组的数目小于 2^n× (2^n−1)/2”，仅使用了一个例子来说明。”的处理

处理方法：在修改后的论文对定理 9 作了更为严密的阐述和证明。

原因：有关CP_nets的完备性的判定定理，算法的构造，一个CP_nets可表达的偏好的个数是多少等本身就是一个重大的课题，即什么情况下CP_nets是完备的，什么情况下CP_nets不是完备的，完备性与CP_nets的结构及其条件偏好表具有什么样的关系等等？这些都还未彻底解决。虽然目前作者完全解决了参考文献[5]的可分离的条件偏好网的完备性问题(即若CP_nets由一些没有依赖关系的孤立的点来组成的化，可我们可求出n个顶点的可分离的CP_nets能表达的偏好的个数是 3^n-2^n，因此当$n > 1$ 时都不完备)，但其他结构的CP_nets所能表达的偏好的个数还正在探索中，任意结构的CP_nets可表达的偏好的个数是多少对应如下的一个图论问题：**按照图的拓扑排序给定的一个图的入度序列，确定该图的入度序列所能表达的偏好的个数m是多少？m与偏好完备性所需要的 $2^n\times(2^n-1)/2$ 相差多少？这是一个有着重大意义的课题。**

当然一个CP_nets图是否完备还可从如下的思路来求取，**能否求出不完备的一些特殊子图的结构形式，看这些子图是否都出现在一些其他的母图中（类似于图论中平面图的库拉图夫斯基判定定理，找母图中作任何同胚操作后不含K_5及$K_{3,3}$子图）**，即我们重点找类似于平面图中的坏的基因K_5和$K_{3,3}$图。那么我们能否找到使得CP_nets图不完备的一些子图呢？幸运的是，我们已经找到了一些（如存在两个度为 1 的顶点的图，即有两个树叶的子图，类似于论文中的图 2 等）。类似于平面图判定中K_5及$K_{3,3}$子图的作用，在图的完备性判定中起着类似作用的子图都有哪些呢？作者还在探索。

综上所述，正是由于图的完备性问题还未彻底解决，所以原论文的一个想法是通过实例引出 CP_nets 在描述偏好会出现不完备的情况，所以只是粗略地描述 CP_nets 的完备性，以使读者了解完备性是表达能力很重要的一个方面。引入定理 9 的作用在于引出问题，促使读者继续探讨什么情况下是完备的，什么情况下是不完备的，不完备的程度如何等。所以在原论文和修改后的论文中的 4.2.2 节都给出了完备性的进一步思考，在原论文和修改后的论文中的最后一节“未来工作中”，都给出了继续探讨 CP_nets 的完备性的问题的描述。

十、关于审稿专家的“ 10.性质 1 的描述不是很严格，应该将“直观”的含义准确的表述出来。”的处理

处理方法： 在修改后的论文中，结合 CP_nets 的图形表示，阐述了“直观”含义，具体表现在性质 1 的证明中。

原因：结合定义 5 给出的CP_nets的图形表示的三个内容：顶点集V，边集CE，和条件偏好表CPT(X_i)，结合性质 1 的证明，很容易看出CP_nets的直观的含义。

十一、关于审稿专家的“11. 在定理 12 中，作者证明了强占优关系的保持特性。反过来，作者没有说明“如果 SCSP 中的解 o1 优于 o2，那么 o1 强占优于 o2。”是否成了。如果上述命题不成立，则不能用测试 SCSP 中解的优劣的方法替代强占优测试。这样，作者声称的贡献（3）就值得商榷了”的处理

处理方法：在原论文和修改后的论文中的摘要，引言及其结论中都指出了从 CP_nets 向 SCSP 的规约，解决了部分 CP_nets 上的强占优测试的高复杂度问题，通过 SCSP 中解的优劣判断来推理 CP_nets 中的强占优测试至少给出了部分解或近似解。

专家的本条建议引发了作者对 CP_nets 的更多思考，并由此产生一些新的研究内容，下面是对专家所提出问题的一些解释。

由于 CP_nets 一般情况下是不完备的，因此可考虑 CP_nets 是完备的情况和不完备的情况。

（1）CP_nets是完备的情况（此时CP_nets为平凡的CP_nets，例如只有一个顶点的CP_nets），此时由于CP_nets中任何两个配置o_1与o_2都可比较，因此要么$o_1 > o_2$，要么$o_2 > o_1$，由于CP_nets向SCSP规约时保持了强占优关系，因此要么$Util(o_1) > Util(o_2)$，要么$Util(o_2) > Util(o_1)$。显然从$Util(o_1) > Util(o_2)$可推理出$o_1 > o_2$。事实上当CP_nets完备时，CP_nets中所有偏好关系的强占优关系与规约后得到的SCSP中的解的优劣关系是一个双射函数，从CP_nets中的强占优关系可得到SCSP中解的优劣关系，从SCSP中解的优劣关系也能得到CP_nets中的强占优关系。

（2）CP_nets 不完备的情况（一般情况下，此时 CP_nets 为非平凡的 CP_nets，例如本论文中图 2 所示的 CP_nets）：

当CP_nets所表达的偏好不完备时，对于任何两个配置o_1与o_2来说，若o_1与o_2可比较（$o_1 > o_2$，或$o_2 > o_1$），则该种情况等同于情况（1），可以用SCSP中解的优劣来代替CP_nets中配置的强占优测试；当配置o_1与o_2不可比较，即通过改进的warshall算法

求出的传递闭包矩阵中o_1 对应的行与o_2对应的列所在的元素值为 0（不具有强占优关系），此时o_1与o_2的强占优关系在CP_nets中没有表达，此时的问题是：在CP_nets没有表达的o_1与o_2的强占优关系本身是不可比较的？还是Agent对二者的关系闭而不谈呢（或CP_nets中本身就无法表达出来呢）？为此又分作如下两种情况。

（2.1）当o_1与o_2的不可比较关系是因为Agent闭而不谈时，或者因为CP_nets无法表达出o_1与o_2的强占优关系时，此种情况类似与（1）情况，此时CP_nets向SCSP规约后，要么$Util(o_1) > Util(o_2)$，要么$Util(o_2) > Util(o_1)$，因此可以推理得到$o_1 > o_2$，或者$o_2 > o_1$。笔者认为这是合理的，因为由于CP_nets表达能力的限制，或者因为Agent不谈o_1与o_2的优劣，系统给出一个优劣顺序也是有道理的。这正如偏好聚合（选举）中，若某些Agent在投票中没有投Agent a的票（通过Approval Vote投票规则，参见参考文献[10]），但集体选举的结果有可能是a获胜。

（2.2）当o_1与o_2的强占优关系本身是不可比较时，由于CP_nets向SCSP规约时，每个配置都有对应的效用值，且不同的配置其效用值也不一样，因此$Util(o_1) > Util(o_2)$，或者$Util(o_2) > Util(o_1)$，此时我们可推理出$o_1 > o_2$，或者$o_2 > o_1$，我们称此种情况为“过规约”，即原先o_1与o_2没有关系，但规约后有了关系。但在CP_nets中，还有大量的具有强占优关系的配置的规约是完全正确的，因此可认为在此种情况下，本文给出的规约技术得到的是一个部分解或近似解。

总之在最坏情况下，本文所设计的规约技术可认为求到的是一个部分解或近似解，在大部分情况下认为是完全解和精确解。问题的核心仍然在于 CP_nets 中没有表达的配置间的强占优关系，到底是具有而没有表达？还是本身就不具有呢？问题的焦点在于 **CP_nets 的不完备性及其不可比较的配置的真实语义是什么？**但这本身在 CP_nets 中目前还没有界定。其实 CP_nets 的魅力就在于不完全信息下的多属性的定性偏好决策，对其没有表达的偏好的处理方法也是一个新的研究方向。这正如人一样，表达了“我爱你”，其含义就是我爱你，表达了“我不爱你”，其含义就是我不爱你，但是当此人并没有表达“我爱你”或“我不爱你”时，谁又能知道他到底是爱你呢？还是不爱你呢？

原因：CP_nets 没有表达出的偏好虽然魅力十足，但如何处理，尤其是其相关的语义解释还是值得研究。但笔者认为在不同的应用背景中，其“配置间的不可比较的语义”也不一样。参考文献[5]在处理 CP_nets 的学习问题时，就考虑了这种不可比较的语义。

十二、关于审稿专家的“其他一些文字错误”的处理

处理方法：专家所提出的文字错误都改正了，并仔细修改了一些原先没有发现的文字错误。

CP_nets及其表达能力研究

作者

CP_nets and It's Expressive Power

Author

Abstract: Preference handling is one of the important researching content in artificial intelligence, whose four studying hotspots are preference representation, preference elicitation, preference aggregation and preference inference. CP_nets (Conditional Preference networks) is a simple and intuitive graphical tool for representing conditional ceteris paribus (all other things being equal) preference statements over the values of a set of variables. But there are very few works that address the problem of expressive power of CP_nets. In this paper, expressive power of CP_nets is studied, in particular, preference completeness, some operations complexity of CP_nets and situation to which model is applicable are discussed detailedly. Firstly, some operations of CP_nets are discussed, by using improved warshall algorithm, we solve the problem of worst case complexity of strong dominance testing with respect to binary-valued CP_nets, and proved its complexity is $O(4^n)$. Secondly, by constructing induced graph of CP_nets and studying its properties, we make a conclusion that CP_nets suits for multiple attributes qualitative decision making and inference under incomplete preference information situation especially. When needing to handle complete preference information, it can be fulfilled by interactive communication with agent. Strong dominance testing is a tractable problem in theory, but it is a intractable problem in practice. In order to solve this exponential order complexity, at last, some reduction techniques from qualitative judging to quantity judging with SCSP (Soft Constraint Satisfaction Problem) are given. Because qualitative judging on c-simiring is linear complexity, it increases the expressive power of some CP-nets. All these works can be seen as the improvement and refinement of Boutilier and Bistarelli's related works.

Key words: conditional preference networks; expressive power; strong dominance testing; preferences completeness; improved warshall algorithm; multiple attribute qualitative decision under incomplete information; soft constraint satisfaction problem

摘　要: 偏好处理是人工智能中的一个重要研究内容, 它的 4 个研究热点是偏好的表示, 提取, 聚合和推理. CP_nets是一种简单直观的偏好表示的图形工具, 但很少有工作在研究CP_nets的表达能力. 本文研究CP_nets的表达能力, 详细研究了CP_nets表达偏好的完备性, 其上构造的运算的复杂度以及适用的场合. 首先给出了CP_nets模型上的几个运算, 利用改进的Warshall算法求出了二值网的强占优测试在最坏情况下的复杂度为$O(4^n)$. 其次通过构造CP_nets导出图及其性质的研究, 得出CP_nets特别适合不完全信息情况下多属性定性偏好决策和推理. 当需要处理更完全信息时, 可借助于与Agent的交互来完成推理任务. 虽然我们给出了CP_nets的强占优测试的理论解, 但其理论上可解, 实际上不可解. 为了解决强占优测试的指数级复杂度问题, 本文最后给出了一种SCSP (带有软约束的满足问题) 的求解方法. 它把CP_nets中的定性运算转为约束半环中的定量运算, 从而将指数级的复杂度转化为多项式的复杂度, 间接提高了部分CP_nets的表达能力. 本文所作的工作是对Boutilier和Bistarelli工作的改进和提高.

关键词: 条件偏好网; 表达能力; 强占优测试; 偏好的完备性; 改进的 Warshall 算法; 不完全信息下的多属性定性偏好决策; 带有软约束的满足问题

中图法分类号: TP301　　　**文献标识码**: A

1　引言

偏好决定着选择，因此在分布式人工智能中，当多个 Agent 相遇并进行合作求解时，理解 Agent 彼此的偏好并推理出彼此对各种事物的感兴趣程度，进而采取有利于合作的方式来进行决策是非常重要的。Agent 的偏好决定了双方合作求解的方式及其速度，因此在人工智能中研究 Agent 的偏好很有意义。

偏好在日常生活中随处可见。我们在购买汽车时，许多人（用 Agent a 来表示）的一个简单准则是便宜的比昂贵的好，同样价钱下，空间大的比空间小的好。而有的人（用 Agent b 来表示）的偏好则不同，他可能首先认为安全的比不安全的好，同样安全系数下，操控灵活的比操控不灵活的好等。因此汽车经销商需要根据客户的偏好描述推理出其对汽车的需求原则，从而

给出客户最需要的一些汽车配置来满足客户的需求。另外偏好在互联网中的搜索引擎也开始应用：一些搜索引擎对搜寻结果的排序，还是按照固定算法来处理并显示，它们很少考虑用户的个性需求（个人偏好）。可以说不同用户对所搜索出的结果的感兴趣程度不一样，如何通过与用户的简单交互（用户选择的网页，及其浏览的时间等），对原先搜索结果重新排序以满足客户的偏好就是一个有意义的课题。这其中牵涉一个重要的理论问题——Agent 偏好的获取。

本文研究偏好处理的基础——偏好表示工具CP_nets[1]的表达能力，属于对CP_nets基本结构和属性的探索，以期为参考文献中的其他研究提供强有力的数学基础。其中第 1 节给出了偏好的研究内容和存在的问题；第 2 节给出了基于CP_nets的偏好处理元模型的几个方面；第 3 节详细讨论基于CP_nets的几个典型运算及其复杂度；基于 2、3 节的内容，紧接着在第 4 节给出了CP_nets的表达能力的描述。为了解决CP_nets在强占优测试上的缺陷，在第 5 节将其规约成受限半环（c-semiring）[2,3]上的解的优劣判断问题，来降低部分CP_nets的求解复杂度，并给出了相关工作的对比。最后一节给出了结论和未来工作。

本文的主要特色和贡献在于：（1）基于CP_nets导出图上的跳变关系所对应的稀疏关系矩阵，利用改进的Warshall算法求出了二值CP_nets上的强占优测试的时间复杂度为$O(4^n)$，解决了JAIR（Journal of Artificial Intelligence Research）[1]上没有解决的最坏情况下的强占优测试的时间复杂度问题，即阐述了强占优测试问题在理论上是可解的；（2）首次从适用的范围、模型的获取、语言上所定义的运算及其复杂度等方面研究了CP_nets的表达能力。指出CP_nets适合不完全备信息下的定性偏好决策。倘若要得到完备的偏好信息时，可通过与Agent的交互计算来完成，这间接提高了CP_nets的表达能力；（3）鉴于CP_nets上的强占优测试的指数复杂度在实践上的不可行性，本文将无环CP_nets上的强占优测试在多项式时间规约为受限半环 [2,3]上的解的优劣判断问题，从而利用SCSP（Soft Constraint Satisfaction Problem，带有软约束的满足问题）来对半环上的数值进行定量运算。由于受限半环上解的优劣判断可在多项式时间完成，因此借助于规约技术，可解决无环CP_nets上的强占优测试的指数复杂度问题。即本文不仅给出了文献[1]没有给出的强占优测试的理论解，还给出了一个实际可行的（复杂度低的）部分解或近似解。

1.1 Agent偏好处理的研究内容

偏好处理是近几年人工智能的研究热点。2004 年发表在世界顶级人工智能期刊JAIR上的Boutilier[1]的论文“CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements”在 2009 年被评为近 5 年最具影响力的一篇论文[1]，因为它开辟了一个新的研究方向。2009 年世界顶级人工智能会议IJCAI2009 上的杰出论文[2]是Koriche[4]的“Learning conditional preference networks with queries”，其内容是有关偏好的学习问题。通过仔细阅读近几年人工智能会议及期刊的一些论文，结合偏好的应用，我们可把Agent的偏好研究归为如下的 4 类。

（1）偏好的表示（Preference Representation）：研究如何将Agent的偏好用数学方式加以描述，并重点探讨这种描述方式的简洁性，易用性及其所用语言的表达能力；（2）偏好的提取（Preference Elicitation）：研究如何通过与Agent交互来学习和推理Agent的偏好。该方向是以机器学习的方式来获取Agent的偏好。其中Koriche[4]提出了利用等价查询（Equivalence Query）和成员查询（Membership Query）来获取Agent的条件偏好。Lang[5]提出了如何学习最简单的一种偏好——属性可分离的偏好（属性之间没有依赖关系，即CP_nets由一些孤立的点组成）。Conitzer[6]提出了利用比较查询的方法来提取Agent的单峰偏好，Chevaleyre[7]给出了多属性域偏好学习的通用准则，给出了哪些实例是可学习的，哪些实例是不能学习的；（3）偏好的聚合（Preference Aggregation）：研究如何将多个Agent偏好聚合成一个集体的偏好，其本质是政治生活中的选举，其核心是设计一种选举制度，使得选举科学、民主、高效和公平。Tang[8]在世界顶级人工智能期刊——Artificial Intelligence上发了一篇文章“Computer-aided proofs of arrow's and other impossibility theorems”，通过程序验证了选举问题中的“阿罗不可能定理”（Arrow Impossible Theorem），即在具有 2 个投票者，3 个被选举人的投票环境中，任何投票规则都会出现独裁者。但该文没有给出投票所应遵循的几个原则之间的相关性问题，且所给的民主性质是否完备也不知道。Lang[9]给出了组合投票和偏好之间的关系，Conitzer[10]给出了在具有少量被选举人的情况下，什么时候操纵选举是困难的问题，其目的是避免选举中的舞弊行为，Walsh[11]给出了在不确定信息下的偏好提取和聚合问题。Zuckerman [12]给出了Agent联盟在选举中的操纵问题，而刘惊雷[13]给出的是Agent联盟在最优联盟结构求解中的应用问题；（4）偏好的推理（Preference Inference）：研究如何从已有的偏好模型和语言中，利用初始的偏好断言及Agent内嵌的推理规则，推理出Agent关于某事物的断言是否一致，是否还有哪些隐含的偏好等。其中Domshlak[14]提出了带有硬限制和软限制的偏好的推理方法，张志政[15]提出并构造了一个能够描述和推理多种类型偏好的逻辑系统MPL(Logic of Many Kinds of Preference)，通过直接引入 4 个逻辑偏好算子构造了能够表示 4 类偏好的MPL语言L_{MPL}，基于包含、交、补 3 个集合操作给出了偏好推理的规律。

在偏好处理的 4 个方面研究中，偏好表示是最基础的工作。目前偏好的应用大多都是基于CP_nets来加以描述的[1−7,14,22−24]。其中最典型的是Boutilier的工作[1]，他对前人工作进行了总结，详细描述了CP_nets的语法、语义及应用。然而该文并没有探讨CP_nets的表达能力，且其上的强占优测试（Strong Dominance Testing）在最坏情况下的算法及其复杂度都没有给出。CP_nets以简洁、直观的图形化方式来表示偏好，近几年出现的一些新的偏好表示方式都是在其基础上产生。例如Brafman[16]就提出了一种带有属性重要性的TCP_net，该模型在处理属性之间的偏好关系基础上，增加了反映属性重要性的元素，然而它的分析技术和 CP_nets的一样。Rossi[17]提出包含多个Agent偏好模型的mCP_net，它实质上是多Agent的偏好聚合模型，因此这些模型都

1 详细的评价内容参见 http://www.jair.org/

2 详细内容参见http://www.ijcai.org

是特殊的一类CP_nets。

1.2 偏好表示存在的问题

偏好处理研究的 4 个方面的基础是偏好表示问题，而偏好表示大多是利用 CP_nets 来表示，虽然对 CP_nets 的研究是近几年国际人工智能界的研究热点，但是 CP_nets 的表达能力如何，即它能表达什么样的偏好，适合解决什么样的问题等，其上运算的复杂度如何等，其表达的偏好是否完备等却很少有人研究。本文从偏好处理的元模型出发，从模型的数学结构，语言的表达能力，运算的复杂度等方面研究 CP_nets 的表达能力，给出它适合描述的问题及适合解决的问题。对于 CP_nets 没有表达出的偏好关系，可借助于与 Agent 的交互来完成。另外还设计了一个可解决任意形状的 CP_nets 的强占优测试算法，给出了一种可得到部分解或部分解的一种规约技术，这些都间接提高了 CP_nets 的表达能力。

2 CP_nets 的语言与语义

2.1 偏好处理的元模型

首先从Brafman [18] 给出图 1 所示的偏好处理的元模型出发，逐步引入CP_nets及其表达能力的相关论述。

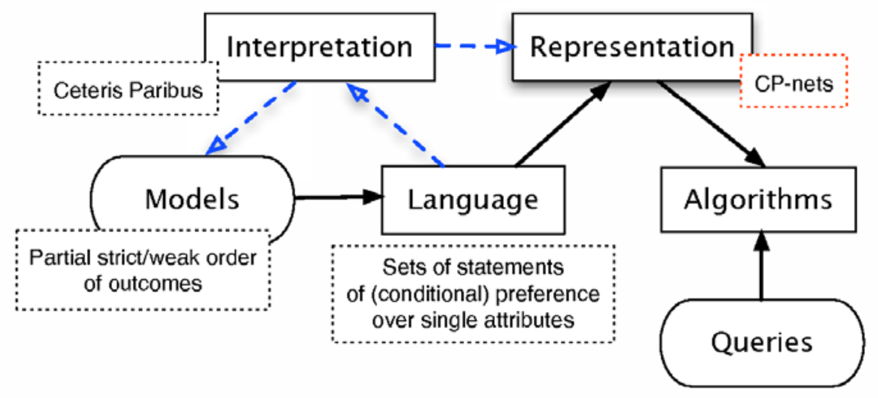


Fig. 1 The meta-model of preference handing

图 1 偏好处理的元模型

偏好处理的元模型主要包含如下元素：偏好的模型（Models）、偏好的语言（Language）、用户对偏好的查询（Query）和实现用户对偏好查询的各种算法（Algorithm）。下面围绕这些方面来描述基于 CP_nets 的偏好处理。

2.2 偏好的模型——严格偏序关系

模型就是一个数学结构，Agent 的偏好是用数学结构——偏序结构来描述。

定义 1. 设V = {X_1, X_2, …, X_n}是决策属性（Attribute）或变量的集合，Dom(X_i)代表属性X_i的有限定义域，则决策空间Ω $=\times_{i=1}^{n} Dom(X_i)$（各个属性定义域的笛卡积）表示所有属性的可能组合。o∈Ω是决策空间的一个配置[1]（Outcome, Alternative），代表决策空间的一种组合。若两个配置o 和o'仅有一个属性值不同，而其他属性值都相同，则称o 和o'为可交换的配置（Swap Outcome）。

现实中决策的两个属性之间可能具有依赖关系，如果Agent对属性X_i的偏好取决于X_j，则称X_j是X_i的一个父亲，用$Para(X_i)$表示X_i的父亲，一个属性可能有多个父亲。

定义 2. ≥是决策空间Ω上的二元关系，

（1）若≥自反（$(\forall o)\ o\in\Omega\rightarrow o \geq o$）、反对称（$(\forall o,o')\ (o, o'\in\Omega \wedge o \geq o' \wedge o' \geq o \rightarrow o = o')$）和传递（$(\forall o, o', o'')\ (o, o', o''\in\Omega \wedge o \geq o' \wedge o' \geq o'' \rightarrow o \geq o'')$），即≥是一个偏序关系时，称≥为Ω上的偏好关系。

（2）若存在两个配置 o，o'∈Ω， $o \not\geq o'$且 $o' \not\geq o$，则称 o 和 o'不可比（Not Comparable）。当Ω中存在不可比的两个配置时，称偏好关系≥不完备。

（3）若对两个配置 o，o'∈Ω有 $o \geq o'$但 $o' \not\geq o$，称 o 和 o'是具有严格偏好关系，可表示为 $o > o'$。

[1] 将Outcome称作配置比较直观．如<P4, 2GDDR3, 320GHD>代表计算机的一种配置.其中CPU是P4,内存是 2G, 硬盘是 320G.

定义 3. 偏好处理的模型 PHM（Preferences Handling Model）是一个数学结构，其形式化为 PHM = ⟨Ω，>⟩，其中Ω是定义 1 中的决策空间，>是定义 2 中严格偏好关系。

2.3 偏好的语言及CP_nets

偏好处理语言是Agent表达其对各个决策属性的偏好断言，在CP_nets中，偏好语言是通过每个决策属性的条件偏好表CPT(X_i)来描述的。

定义 4. 设CPT(X_i)为属性X_i的条件偏好表（Condition Preference Table），它表示属性X_i在其父属性$Para(X_i)$的不同取值下，Agent对Dom(X_i)集合的一个严格偏好。之所以称CPT(X_i)为条件偏好表，是因为$Para(X_i)$在不同取值下，Agent对属性X_i的几个取值的偏好排序也不同。在$Para(X_i)$的所有取值下，Agent对属性X_i的取值的严格偏好排序构成了属性X_i的条件偏好表CPT(X_i)。

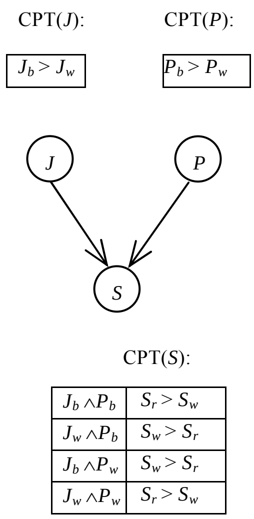
定义 5. CP_nets是一个有向图N = <V, CE >，其中V是顶点集（如定义 1 所示），CE为有向边集，代表所有属性之间的依赖关系。即一条有向边起点的取值影响着终点取值的偏好（边的源点取值确定了终点不同值的一种排序）。对于每一个顶点X_i，都有一个条件偏好表CPT(X_i)与其关联。

定义 6. 偏好处理的语言 PHL（Preferences Handling Language）是 Agent 在每个决策变量 X 上的条件断言的并，即 PHL $=\bigcup_{X_i \in V} \mathrm{CPT}(X_i)$。

为进一步理解CP_nets的语法、语义及对其进行的各种操作，下面给出一个典型的“晚会穿衣”实例[1]。本文所有概念、性质与定理都可通过它来理解。

例 1: 某人参加晚会的穿衣主要考虑夹克（Jacket），裤子（Pants）和衬衫（Shirt），分别用 J，P 和 S 来代表。对于夹克和裤子来说，他无条件喜欢黑色，而不是白色。而穿什么颜色的衬衫取决于夹克和裤子颜色的组合。如果夹克和裤子是同一种颜色，则他偏好红色衬衫而不是白色衬衫，当夹克和裤子颜色不同时，则偏好白色衬衫而不是红色衬衫。

该例子的CP_net图N = <V, CE >如图 2 所示。其中V = $\{J, P, S\}$，Dom (J) =$\{J_b, J_w\}$[1]，Dom (P) = $\{P_b, P_w\}$，Dom (S) =$\{S_r, S_w\}$[2]；CE = $\{<J, S>, <P, S>\}$。各个顶点的偏好表在图 2 中。



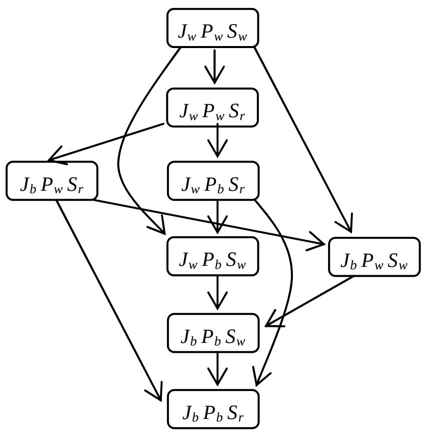


Fig. 2 CP-nets for “Evening Dress"
图 2 “晚会穿衣”的 CP_nets

Fig.3 The induced graph of CP_nets
图 3 CP_nets 的导出图

2.4 偏好语言的语义

Agent给出了其条件偏好语言CPT(X_i)后，如何将其映射或解释为偏好处理的模型呢？为此引入Ceteris Paribus语义，利用它Agent不用反省（不用思考，下意识地）就可得到其含义。

基于CP_nets的偏好遵循Ceteris Paribus（all else being equal）语义[1]，即对于一个配置$o \in \Omega$来说，主要关心除某个属性X_i的值不同外，而其他属性值都相同的话，Agent对属性X_i不同取值的一种偏好断言。利用Ceteris Paribus语义，很容易对可交换的两个配置快速给出偏好断言。因为可交换的两个配置的对比本质上是一个属性值的对比。这正如对两台电脑的配置作对比时，若两台电脑只有一项配置不同（一个属性值不同），而其他配置都一样的话（其他属性值都一样），任何人不用思考就可给出谁优谁劣的判断。

例 2: 考虑例 1 的“晚会穿衣”实例，“j, p：$s_1 > s_2$”断言可用Ceteris Paribus语义解释为：当其他所有的属性值都一样的话，

1 J_b 代表黑色夹克, J_w代表白色夹克

2 S_r 代表红色衬衫, S_w代表白色衬衫

若$J = j$，$P = p$时，则Agent对$S = s_1$的偏好大于对$S = s_2$的偏好。显然从图 2 中CPT(S)的第 1 行可知，j为J_b，p为P_b时，s_1为S_r，s_2为S_w。

与 Ceteris Paribus 语义有关的概念是如下的条件偏好无关。

定义 7. 设X，Y和Z是Agent的决策属性V的一个划分[19]（即$X \cap Y = \varnothing$，$X \cap Z = \varnothing$，$Y \cap Z = \varnothing$，$X \cup Y \cup Z = V$），对于Z的一个赋值z和任意的x_1，$x_2 \in$Dom(X)，任意的$y \in$Dom(Y)，$x_1\, y\, z > x_2\, y\, z$，蕴含着任意的$y' \in$Dom(Y)，$x_1\, y' z > x_2\, y' z$，则称属性集X与属性集Y在条件Z的一个赋值z下偏好无关[1]。

例 3: 假定一个CP_nets图N = <V，CE>，其中V = {A, B, C, D, E}，CE = {<A,C>, <B,C>, <C,D>, <D,E>}，Dom(A) = {a_1, a_0}，Dom(B) = {b_1, b_0}，Dom(C) = {c_1, c_0}，Dom(D) = {d_1, d_0}，Dom(E) = {e_1, e_0}，CPT(A) =“$a_1 > a_0$”，CPT(B) =“$b_1 > b_0$”，CPT(C) =“$(a_1 \wedge b_1) \vee (a_0 \wedge b_0)$：$c_1 > c_0$，$(a_1 \wedge b_0) \vee (a_0 \wedge b_1)$：$c_0 > c_1$”，CPT(D) =“$c_1$：$d_1 > d_0$，$c_0$：$d_0 > d_1$”，CPT(E) =“$d_1$：$e_0 > e_1$，$d_0$：$e_1 > e_0$”。

则属性集{D}与属性集{A, B, E}在C的一个赋值（C = c_1或c_0）下条件偏好无关；同样属性集{E}与属性集{A, B, C}在D的一个赋值（D = d_1或d_0）下条件偏好无关；属性集{C}与属性集{D, E}在{A,B}的一个赋值（AB = a_1b_1或a_1b_0或a_0b_1或a_0b_0）下条件偏好无关。

定理 1. 设N = <V, CE >是CP_nets，对于可交换的两个配置o_1与o_2，则Agent 对o_1与o_2的偏好比较可在线性时间内O(n)内完成，即在$O(n)$内确定出$o_1 > o_2$，或者$o_2 > o_1$。

证明：根据定义 1 可知，可交换的两个配置o_1与o_2只有一个属性值不一样，首先可在线性时间内确定出哪个属性值不一样，而其他属性值都一样（实质是比较o_1与o_2首个不相同的字符，而后确定其对应的决策属性）；当确定出值不同的单属性变量X_i后，求出该属性的父亲$Para(X_i)$，此时o_1与o_2偏好强弱的判断就是CPT(X_i)中的一条记录了，很显然从X_i的条件偏好表中就可确定Agent对o_1与o_2的偏好强弱。

例如例 1 中，若o_1 =“$J_b\, P_b\, S_w$”，o_2 =“$J_b\, P_b\, S_r$”，则在线性时间可确定o_1与o_2在属性S处的值不一样；可求出S的父亲是{J, P}，因此o_1与o_2的偏好强弱对比就是在CPT(S)中寻找“$J_b\, P_b$”这一行的过程，显然在第 1 行包含断言“$J_b \wedge P_b$：$S_r > S_w$”，因此Agent对o_2的偏好大于对o_1的偏好，即$o_2 > o_1$。

3 CP_nets 上的几种运算

利用 CP_nets 可对 Agent 的偏好进行描述，在 CP_nets 上定义一些运算，就成为实用的偏好描述语言，且可对该模型所能表达的配置进行优劣比较。其实图 1 给出的元模型中的“Algorithm”就是指该模型给用户提供了那些算法，以满足用户对模型查询的需要。要想理解这些算法的实现及其复杂度的求解，首先给出 CP_nets 的导出图。

3.1 CP_nets的导出图及强占优测试

3.1.1 CP_nets 的导出图及性质

根据定理 1，Agent 可快速在可交换的两个配置间作出偏好判断。那么如何对任意两个配置之间的强占优关系进行判断呢？为此引入 CP_nets 的导出图及其强占优测试。

定义 8. 设 N = ⟨V，CE⟩是一个 CP_nets，则有向图 N' = ⟨Ω，IE⟩是 N 的导出图，其中Ω如定义 1 所示，IE 是可交换的配置所构成的有向边集，且 Agent 对有向边终点的偏好大于对有向边起点的偏好。

定理 2. N' = <Ω，IE >是二值无环CP_nets图N的导出图，即在N中|V| = n，|Dom (X_i)| = 2，则

（1）$|\Omega| = 2^n$，$|IE| = n*2^{n-1}$。即N'中顶点个数是 2^n，边的个数是$n*2^{n-1}$。

（2）$\forall o \in \Omega$，$Degree(o) = n$，即每个顶点的度都是 n。

证明：（1）先证$|\Omega| = 2^n$。因为$|Dom(X_i)| = 2$，而$\Omega = \times_{i=1}^{n} Dom(X_i) = Dom(X_1) \times Dom(X_2) \times \ldots \times Dom(X_n)$，因此$|\Omega| = |Dom(X_1)| \times |Dom(X_2)| \times \ldots \times |Dom(X_n)| = 2^n$。

再证$|IE| = n*2^{n-1}$。对于任意一个配置o来说，由于o中包含n个属性，且每个属性有两种取值，故与o可交换的配置有n个，另外Ω中共有 2^n个配置，而每个配置有n个可交换配置，因此总共可交换的配置是$n*2^n$个。但由于可交换的配置具有对称性，而N'中的边为有向边，两个可交换的配置在N'中仅有一条有向边相连，故总共边数是$n*2^n/2 = n*2^{n-1}$。

（2）因为|V| = n，即一个配置 o 含有 n 个属性，o 共有 n 个可交换的配置，所以 o 的度为 n。

例如图 3 就是图 2 的导出图，其中顶点个数是 8，边的个数是 12 （$n*2^{n-1} = 3*2^{3-1}$），每个顶点的度都是 3，满足定理 2。

3.1.2 跳变关系及强占优测试

定义 9. 设 N' = <Ω，IE >是 CP_net 图 N 的导出图，对于 o，$o' \in \Omega$，若从顶点 o'到顶点 o 可达，即存在着一条路径连接顶点 o'和 o，则称 o 强占优 o'，记作 $N \vDash o > o'$。判断 $N \vDash o > o'$是否成立的测试称作强占优测试。

定义 10. 若两个配置 o 和 o'可交换，且 $N \vDash o > o'$，则称 o'和 o 具有跳变关系 FR（Flip Relation），即 o' FR o。

事实上，跳变关系 FR 就是导出图 N'中的边集 IE，即 FR = IE。

跳变关系仅仅给出了可交换的配置之间的关系，具有跳变关系的两个配置具有强占优关系，但不是可交换的配置也有可能具有强占优关系。由于强占优关系具有传递性，因此对 FR 或 IE 关系求传递闭包，即可求出所有的强占优关系。

3.1.3 CP_nets 的强占优测试算法

沃舍尔（Warshall）算法是基于关系矩阵运算而得到关系传递闭包的一个算法[19]，当关系矩阵是$m\times m$的矩阵时，其时间复杂度为$O(m^3)$。由于CP_nets导出图N的关系矩阵是稀疏矩阵，我们可进一步将其复杂度降低为$O(m^2)$。改进的沃舍尔算法Improved-Warshall 的完整描述如图 4 所示，可认为它是CP_nets的强占优测试算法。

```
Input : CP_nets的导出图N' = <Ω, IE>中IE的关系矩阵A[m][m]，其中m = 2^n
Output : IE 关系的传递闭包的关系矩阵 B[m][m]

Step1：//求矩阵 A 的所有行中非零元素所在的列号于数组 C
 For i =1 to m   Do
   For j =1 to m   Do
     If  ( j ∈A[i, j] )   Then C[i]: =C[i] ∪ { j }
     //将关系矩阵的i行为1的列号保存在数组C[i]中

Step2：//把 C[j]中出现的元素所在的集合都并到 C[j]中,
 For i =1 to m   Do
   For j =1 to m   Do
     If  (i∈C[j] and ( j≠ i ) ) Then C[j]: =C[j] ∪ C[ i ]
     //把C[j]中的元素所在的集合并到C[j]中
     //其本质是把第i行加到第j行为1的元素所在的那些行中

Step3：//根据数组 C 的值设置传递闭包矩阵 B
 For i =1 to m   Do
   For j =1 to m   Do
     If  (j∈C[i]) and (B[i, j]= 0) )
       Then B[i, j]: = 1   //将顶点i和顶点j有强占优关系，则置B矩阵的i行j列为1
```

Fig. 4 Improved warshall algorithm for strong dominance testing

图 4 求强占优测试的改进的 Warshall 算法

例 4: 利用改进的Warshall算法求图 3 导出图N' = <Ω，IE >的传递闭包。其中$\Omega = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8\}$，$o_1$ =“$J_w P_w S_w$”，o_2 =“$J_w P_w S_r$”，o_3 =“$J_w P_b S_r$”，o_4 =“$J_w P_b S_w$”，o_5 =“$J_b P_b S_w$”，o_6 =“$J_b P_b S_r$”，o_7 =“$J_b P_w S_r$”，o_8 =“$J_b P_w S_w$”；IE = {<o_1, o_2>, <o_2, o_3>, <o_3, o_4>, <o_4, o_5>, <o_5, o_6>, <o_1, o_4>, <o_1, o_8>, <o_2, o_7>, <o_3, o_6>, <o_7, o_6>, <o_7, o_8>, <o_8, o_5>}。

解： IE 的关系矩阵 A 为：

$$\begin{matrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix},$$

（1）根据 Improved Warshall 算法的描述，当执行完 Step1 的双层循环后，数组 C 的数值如下：

C[1] = {2, 4, 8}，C[2] = {3, 7}，C[3] = {4, 6}，C[4] = {5}，C[5] = {6}，C[6] = ∅，C[7] = {6, 8}，C[8] = {5}。

（2）算法的 Step2 的执行步骤较多，执行完后数组 C 的数值为：

C[1] = {2, 3, 4, 5,6,7,8}，C[2] = {3,4,5,6,7,8}，C[3] ={4, 5, 6}，C[4] ={5,6}，C[5] = {6}，C[6] = ∅，C[7] = {5,6, 8}，C[8] = {5}。

（3）算法的 Step3 执行完后，求出的传递闭包矩阵 B 为：

$$\begin{matrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix}$$。

此时传递闭包矩阵 B 中元素 1 的个数是 23，代表 CP_nets 可表达的强占优个数为 23。而图 3 中总共应该有 8*7/2 = 28 对可比较的配置，但利用图 2 的 CP_nets 只表达了 23 对强占优关系，其余 5 对没有作阐述，即图 2 的 CP_nets 对偏好的表示不完备。

3.2 弱占优测试

从 3.1 节可知，CP_nets 所能表达的强占优关系有限，当 CP_nets 所表达的偏好不完备时，如何增强 CP_nets 的表达能力呢？为此引入弱占优测试。

定义 11. 设 N' = <Ω，IE >是无环 CP_nets 图 N 的导出图， 若从顶点 o 出发的所有路径都不能到达 o'，（$o' \not\succ o$），则称 o 弱占优 o'，记作 $N \nvDash o' > o$。判断 $N \nvDash o' > o$ 是否成立的算法称作弱占优测试。

o 弱占优 o'的含义是 o' 不比 o 强。显然 o 强占优 o'，则 o'不可能强占优 o。但 o 弱占优 o'，则 o'也可能弱占优 o。弱占优关系的作用在于对配置进行拓扑排序。当 $N \nvDash o' > o$ 时，则对不同配置的产品排序时，o 可以排在 o'的前面。换句话说，在关于 Agent 的偏好断言中，没有足够的信息确定 o 和 o'谁强谁弱，但可确定谁不比谁强。

定理 3. 设 N 是无环 CP_nets，$\forall\, o, o' \in \Omega$，$N \vDash o > o'$ 是 $N \nvDash o' > o$ 的充分条件，但不一定是必要条件。

证明：显然 $N \vDash o > o'$说明在 N 的导出图 N'中，存在着一条从 o'到 o 的路径，由于 N 的无环，则不存在一条从 o 到 o'的路径，$N \nvDash o' > o$；但 $N \nvDash o' > o$ 也可能表示 o' 和 o 不可比较，因此 $o > o'$也可能不成立，即由 $N \nvDash o' > o$ 得不出 $N \vDash o > o'$。

定理 4. N是无环CP_nets，o和o'是Ω中两个不同的配置，若在N中存在一个属性X，使得o和o'的属性X的所有祖先值[19]相同，即o[Ance (X)] = o'[Ance (X)]，但o[X] > o'[X]，则$N \nvDash o' > o$。其中Ance (X)表示属性X的所有祖先。[1]

证明：通过构造法来证明。由于N是无环图，因此存在一个顶点Y_1，其入度为 0（可看作根顶点）。然后由CPT(Y_1)比较Agent对$o[Y_1]$ 和$o'[Y_1]$的偏好大小，若$o[Y_1] > o'[Y_1]$，即o在Y_1属性上的值优于o'在Y_1属性上的值，则o'不会比o强，即$N \nvDash o' > o$；同理若$o'[Y_1] > o[Y_1]$，则$N \nvDash o > o'$；若$o[Y_1] = o'[Y_1]$，则在N中删除顶点Y_1，形成新的CP_nets图N_1，而后在N_1中寻找根顶点Y_2，再根据CPT(Y_2)比较Agent对$o[Y_2]$ 和$o'[Y_2]$的偏好大小，若$o[Y_2] > o'[Y_2]$，同理得到$N \nvDash o' > o$；若$o'[Y_2] > o[Y_2]$，则$N \nvDash o > o'$；若$o[Y_2] = o'[Y_2]$，则在N中删除顶点Y_2，形成新的CP_nets图N_2，…，依次类推。由于o和o'是不同的配置，因此必定能找到首个不相同的属性值，并根据条件偏好表判断该属性值的优劣。

例 5: 考虑例 3 中的CP_nets，假设配置o_1 = "$a_1\, b_1\, c_1\, d_1\, e_1$"，$o_2$ = "$a_1\, b_1\, c_0\, d_1\, e_0$"，$o_3$ = "$a_1\, b_1\, c_0\, d_0\, e_1$"，由于$o_1$[Ance (C)] = o_2[Ance (C)] = "$a_1\, b_1$"，而从CPT(C)可知$o_1[C] > o_2[C]$（因为$o_1[C] = c_1$，$o_2[C] = c_0$，在C的条件偏好表中有一行记录：$(a_1 \wedge b_1)$: $c_1 > c_0$），根据定理 4 有$N \nvDash o_2 > o_1$。同理有$N \nvDash o_2 > o_3$。

基于定理 4 可给出图 5 所示的 CP_nets 的弱占优测试算法 Algorithm1，它是基于 CP_nets 图 N 的搜索算法。

Input **:** CP_nets 图 N = <V, CE>和配置 o，$o' \in \Omega$
Output **:** $N \nvDash o > o'$或者 $N \nvDash o' > o$

[1] 证明思路是：对N进行拓扑排序后得到顶点序列L，然后从L的首个位置开始逐个比较o和o' 在对应位置处的属性值的偏好强弱。简单来说，若o的首个属性值强于o' 对应位置处的属性值，则o弱占优o'（o' 不会比o强）。

```
U = V
While (U != NULL)    Do
{
     Y = GetRoot(U) //取得 U 的根顶点
     If (o[Y] > o'[Y])    N ⊭ o' > o    break
     If (o'[Y] > o[Y])    N ⊭ o > o'    break
     If (o[Y] = o'[Y])    U = U−Y
} End While
```

Fig. 5 weak dominance algorithm Algorithm1

图 5 弱占优判断算法 Algorithm1

定理 5.

（1）二值CP_nets图N的强占优测试（$N \vDash o > o'$）算法的时间复杂度为$O(4^n)$；

（2）二值无环 CP_nets 图 N 的弱占优测试（$N \nvDash o' > o$）算法的时间复杂度是 $O(n)$。

证明：（1）因为改进的Warshall算法可在$O(m^2)$时间内求出关系IE的传递闭包，其中m为关系图中顶点的个数。而算法输入是CP_nets的导出图N'，由于N'中的顶点个数是 2^n，此时$m = 2^n$，故利用改进的Warshall算法可在$O(4^n)$求出其传递闭包，倘若配置$<o, o'> \in IE^*$，则o'强占优o。

（2）略。

3.3 基于弱占优的拓扑排序

现实中的决策空间有指数级个配置，希望Agent对任何两个配置作出强占优判断是不可能的，因为Agent需要进行 $2^n \times (2^n - 1)/2$ 次判断。倘若$n = 10$，则进行约 50 万次的偏好判断，这显然不可能。另外即使有足够的时间来决策，但在大量的两个配置之间，尤其是两个配置之间有多个属性值不一样的话，Agent会面临“鱼和熊掌不可兼得的局面”，他会很难做出快速而又一致的判断。但基于CP_nets的Ceteris Paribus语义，Agent可进行条件偏好无关的断言，而后利用弱占优测试来对感兴趣的一些配置进行排序。

由于偏好是一种偏序，因此对所有配置进行全序排列不可能，为此我们可使用拓扑排序。另外虽然决策空间Ω很大，但现实中往往只关心其一个子集 O⊆Ω。例如捷达轿车在大众 4S 店仅有十种左右配置（其他配置的不生产），客户购买汽车时也只是在 O 子空间中进行决策，为此我们可基于弱占优查询来实现 O 的拓扑排序。O 中配置进行拓扑排序后，就可找出最优配置和最差配置了。

定理 6. 设 O⊆Ω是 Agent 感兴趣的配置集，则基于弱占优测试可对 O 进行拓扑排序，且拓扑排序的复杂度为 $O(n*pLog(p))$。其中 p=|O|为感兴趣的配置的个数。

证明：显然 $N \nvDash o' > o$ 时，o 可排在 o'的前面。利用快速排序(Quick Sort)算法可在|O|Log(|O|)时间内将|O|个配置排好序，而排序所进行的比较操作（弱占优测试）需进行|V| = n 次，因此拓扑排序操作的次数是|V|*|O|*Log(|O|) = $n * p\,Log(p)$，故拓扑排序的复杂度为 $O(n*pLog(p))$。

定理 6 表明，基于弱占优测试对感兴趣的配置进行排序的时间复杂度不高于多项式时间复杂度，因此其在实践中可行。

4 CP_nets 的表达能力

4.1 表达能力的衡量方法

基于模型产生的语言的表达能力如何，直接决定了用户使用该语言所能解决的问题的能力和复杂度。那么我们希望该语言具有什么样的特性，即什么样的语言的表达能力较强？为此我们给出判断偏好表达能力强弱的如下标准：

（1）模型是否简洁而又清楚地表达 Agent 的偏好（即能对偏好说明白吗）。

（2）给出的偏好模型是否让别的 Agent 不用反省就能听明白（即能听明白 Agent 表达的偏好吗）。

（3）语言中是否有足够的算法来实现用户对偏好的各种查询（即提供偏好处理的算法充分吗）。

（4）实现偏好测试的算法的可行性如何（即时间复杂度大吗）。

（5）模型可表达的断言和实际中的断言数量相差多少（即表达的偏好完备吗）。

（6）偏好语言的适用范围是否宽泛（即适合解决一个还是一类偏好的处理问题）。

4.2 CP_nets的表达能力

基于 4.1 节表达能力的 6 个衡量方法，下面依次给出基于 CP_nets 的偏好表达能力的对应描述和相应性质与定理。

（1）对于外行用户来说，利用CP_nets很容易表达它对事物的偏好断言，即直观性很强。因为CP_nets中的CPT(X_i)是Agent的偏好表示语言，其对偏好的断言基于单属性X_i值不同而其他属性值都相同的配置的偏好比较。所以利用CP_nets，Agent能正

确而直观地表达自己的偏好，即基于CP_nets，Agent能将其欲表达的偏好说明白。

（2）对于可交换的两个配置来说，Agent 可在线性时间内得出其强占优关系，因此别的 Agent 能听明白两个可交换配置的强占优关系，对于任意两个配置来说，也可在线性时间内得出其弱占优关系。倘若要对任意两个配置确定其强占优关系，则基于 CP_nets 的偏好表示却需指数级的时间内才可得到。

即使强占优测试的复杂度很高，但还有些配置之间的强占优关系本身就没有表达出来（不可比的配置，从例 3 可知）。因此基于 CP_nets 的偏好表示，听者能部分听清楚、听明白（判断可交换的配置间的强占优关系）；还有大部分能听明白（导出图的传递闭包包含了大多数的强占优关系），但听起来费劲（求传递闭包需指数时间复杂度），还有部分偏好听不明白（部分强占优关系 CP_nets 没有表达），因为 CP_nets 的模型本身就没有表达。

（3）基于 CP_nets 的偏好表示可实现的算法有强占优测试，弱占优测试，拓扑排序，求最优配置，最差配置（拓扑排序后的第 1 个和最后 1 个元素）。这对于不完全信息下的偏好处理是充分的，即基于这些算法，Agent 可查询得到配置间的任何比较信息。

（4）CP_nets 上的算法除强占优测试的复杂度很高外，其他算法都是多项式时间复杂度。因此基于 CP_nets 的偏好表示中的算法在理论上是可行的（都能求出时间复杂度），在实践中大部分是可行的（时间复杂度是多项式级的）。

（5）CP_nets 能表达的偏好关系就是 N 的导出图 N'所能表达的信息。从例 3 可看出 CP_nets 所表达的偏好信息不完备。但是 CP_nets 理论上所能表达的偏好信息和实际应该表达的偏好信息相差多少，CP_nets 图 N 的结构与其所表达的偏好关系是什么，这还是一个悬而未决的问题，也是后续工作继续探求其表达能力所努力的方向。

（6）CP_nets的适用范围比较宽泛，尤其适合于定性条件下的偏好处理。由于现实中的许多偏好难以定量表达[1]，这为利用CP_nets处理定性偏好提供了舞台。

将上述 6 条表达能力的直观描述加以总结，可到达如下关于 CP_nets 表达能力的两个性质和四个定理。

性质 1. （偏好表达的直观性）：利用 CP_nets 图来表达 Agent 对配置的偏好是直观的（Intuitive）。

证明：根据定义 5 及图 2 给出的一个实例可知，CP_nets是一种图形化的偏好表示工具，其对配置的偏好表示的直观性体现在如下 3 点：（1）顶点集V是决策的变量集，因此从V可知配置包含哪些决策变量；（2）边集确定了决策变量之间的依赖关系，一个顶点的父亲集合决定了该顶点的偏好，例如图 2 中J和P的不同取值，确定了对S的偏好；（3）利用条件偏好表容易表达可交换配置之间的强占优关系。对于例 1 来说，顶点J的条件偏好表中有一个条目为"$J_b > J_w$"，因此根据Ceteris Paribus语义，可产生如下 4 对可交换配置之间的强占优关系$J_b\,P_b\,S_r > J_w\,P_b\,S_r$，$J_b\,P_b\,S_w > J_w\,P_b\,S_w$，$J_b\,P_w\,S_r > J_w\,P_w\,S_r$，$J_b\,P_w\,S_w > J_w\,P_w\,S_w$。同样很容易根据其它顶点的条件偏好表生成可交换配置的强占优关系。

性质 2. （偏好理解的部分简易性）：利用 CP_nets 图 N 来表达的 Agent 的偏好，对于具有可交换关系的两个配置的理解很简单，其可在 $O(n)$时间内可求出其强占优关系。

定理 7. （弱占优运算的充分性）：CP_nets 图 N 上的弱占优查询可表达决策空间中的所有弱占优关系。

证明：Agent 可完成偏好的比较，求拓扑排序，最优配置，求最差配置等其他运算，且这些运算表达了决策空间中的所有弱占优关系，因此弱占优运算是充分性。

定理 8. （弱占优运算的有效性）：在不完全信息下，基于弱占优查询而实现的偏好比较，拓扑排序，求最优配置，求最差配置的运算能在多项式时间完成，因此算法是有效的。

证明：由 3.2 和 3.3 节的定理 5 和定理 6 很容易得出结论。

定理 9. （强占优关系的不完备性）：利用 CP_nets 图 N 来表达 Agent 对配置间的强占优关系一般情况下是不完备的，即存在某个 CP_nets 图 N，其所表达的强占优关系是不完备的。

证明：通过构造反例来证明。对于n个顶点的二值CP_nets图，其决策空间有 2^n个配置，它可以表达的强占优关系图应该是2^n个配置组成的单向完全图，即图中顶点数目是 $2^n\times(2^n-1)/2$。但从例 3 可知（此时$n = 3$），此时该CP_nets导出图求了传递闭包后元组的数目为 23，小于 3 个顶点的CP_nets图应该可表达的偏好的个数是 28（$2^n\times (2^n-1)/2 =2^3\times (2^3-1)/2$），即该CP_nets图可表达的强占优关系的数目比其应该表达的数目要少，因此存在某些配置，其强占优关系在该CP_nets上是无法表达的（例如配置$J_b\,P_w\,S_r$和$J_w\,P_b\,S_r$是不可比的）。因此存在某个CP_nets，其所表达的偏好不完备。

定理 10. （适用范围的宽泛性）：在不完全信息下，利用 CP_nets 图 N 可实现多属性的定性偏好处理。

证明：CP_nets图N中的条件偏好表CPT(X_i)中每一个记录是对X_i不同取值的一个排序，即它表达的是定性关系，CPT(X_i)中的"$x_1\,y\,z > x_2\,y\,z$"表达了对$x_1\,y\,z$的偏好大于对$x_2\,y\,z$的偏好，但并没有表达具体的偏好值是多少，也没有表达$x_1\,y\,z$的偏好比$x_2\,y\,z$的偏好大多少。结合定理 9 所阐述的某些CP_nets表达的偏好不完备，则可达到CP_nets适合于不完全信息下的多属性定性偏好处理，且它对定性偏好的表示是宽泛的。□

事实上，偏好包含两类，一类是定量偏好（Quantitative Preference），一类是定性偏好（Qualitative Preference）[24]。其中定量偏好应用在古典的决策论和决策分析中，且采用的概念是效用函数（Utility Function）。效用函数的本质就是货币化，即现实中的产品虽然很多，但一经效用函数货币化后，每个产品就有了一个价格，而根据价格大小就可确定对不同产品的偏好。而这和现实中许多人对名表名首饰的偏好是一样的，他之所以偏好的程度比较大，是因为该产品的效用函数比较大（价格高，值钱）。但是现实中还有大量的定性偏好，人们去寻求它的效用函数时很困难或根本不可能时（例如人们对同一牌子汽车的不同配置的偏好是不一样的，但人又很难说出两种不同配置的效用函数是多少，这正如说，我爱你爱的很深，但又很难说出我爱你到底有

几分），此时我们可借助于定性偏好的表示方法来研究，而CP_nets恰好就是一种定性偏好的表示语言，因此利用CP_nets可实现多属性的定性偏好处理。

从定理 9 可知，某些 CP_nets 可表达的配置间的强占优关系不完备。CP_nets 的结构不同，其所能表达的强占优的数量也不同，但总共能表达多少个强占优关系，即传递闭包中的元组中的数目是多少，是下一步研究的一个课题。因为它的解决，就可确定基于 CP_nets 的偏好推理与查询哪些是准确查询，哪些是模糊与近似推理。而定理 10 阐述了 CP_nets 可对不完全信息下的多属性下的定性偏好进行处理。虽然在决策空间中存在一些强占优关系在 CP_nets 中没有表达，但借助于 3.2 节的弱占优查询和 4.3 节的与 Agent 之间的交互，可间接处理 CP_nets 没有表达的一些偏好关系。所以 4.3 节描述的是如何提高 CP_nets 处理能力的一个方法。

4.3 与Agent交互实现更强的处理能力

基于CP_nets的偏好处理是经典的基于字符的计算，近几年还出现了一些新型的计算——网格计算和云计算等。与经典计算相比，所有新型计算的本质特点是交互，交互一方面是求解问题复杂的源泉，另一方面又是处理复杂问题的有效手段。计算和交互互为表里：计算是交互的外部表现，交互是计算的内部实现[20]。计算和交互的统一理论是计算机科学的基础，以支撑计算机科学的整套理论体系。

虽然CP_nets可表达的强占优关系不完备，且强占优测试是指数级复杂度，但幸运的是CP_nets上的弱占优测试是充分的，且该运算是多项式时间复杂度，因此基于弱占优测试可实现决策空间的拓扑排序，求极大元，求极小元[19]等。但最大（小）元却不一定能求出，因为最大元强占优任何元，而CP_nets中有的强占优关系没有表达。而根据定理 9 阐述的强占优关系若不完备的话，则这些极大（小）元的强占优关系的判断无法从CP_nets中获得，为此产生和Agent的交互，将多个配置孰优孰劣的判断交由Agent自己决策。此时偏好断言只是个体的事情，与Agent系统无关。下面给出提高CP_nets表达能力的算法Algorithm2。

Input **:** CP_nets 图 N = <V, CE>
Output **:** N 中的强占优序列

1 基于弱占优测试算法求出CP_nets图N的极大元集$N_{Maximum}=\{ o_1, o_2, \ldots, o_m \}$，其中$m$是极大元的个数（即$|N_{Maximum}| = m$）
2 基于强占优测试算法Improved-Warshall判断$N_{Maximum}$中是否存在最大元和最小元
  2.1 ***If*** 存在最大元o_{max}，则将o_{max}排在最前面
  2.2 ***If*** 存在最小元o_{min}，则将o_{min}排在最后面
3 将除了最大元和最小元的极大元集$N_{Maximum}-\{o_{max}, o_{min}\}$返回给提供CP_nets图N 的Agent进行交互计算
  3.1 对$N_{Maximum}-\{o_{max}, o_{min}\}$中的配置进行拓扑排序，得到序列$P_{N'}$
  3.2 将最大元o_{max}排在$P_{N'}$的最前面，最小元o_{min}排在$P_{N'}$的最后面，得到交互计算后的拓扑排序序列$<o_{max}, P_{N'}, o_{min}>$
4. Agent根据最终得到的强占优序列$<o_{max}, P_{N'}, o_{min}>$进行决策

Fig. 6 Strong dominance algorithm Algorithm2 by interactive computing
图 6 借助交互计算实现完备的强占优判断算法 Algorithm2

定理 11. （弱占优和交互计算结合的完备性）：基于弱占优运算和 Agent 的交互计算的算法 Algorithm2 对于极大元的强占优关系的求取是完备的。

证明：显然基于弱占优运算可在多项式时间得到所有的极大元集合，倘若决策空间Ω中存在最大元o_{max}和最小元o_{min}，其必定可通过强占优算法得到。对于$N_{Maximum}-\{o_{max}, o_{min}\}$中的元素来说，他们之间的强占优关系在CP_nets图中没有表达，而通过与Agent交互，可以得到$N_{Maximum}-\{o_{max}, o_{min}\}$配置间的强占优关系。

5 利用 SCSP 增强部分 CP_nets 的表达能力

从 3、4 节的描述可知，基于 CP_nets 的偏好处理有如下缺陷：（1）并不一定能将所有配置间的偏好都表达出来；（2）强占优测试的复杂度大。其中缺陷 1 可通过 4.3 节与 Agent 交互来解决，缺点 2 可通过该节采用的规约技术来得到部分解决。因此 4.3 节和该节都是增强 CP_nets 表达能力的一些方法。

5.1 SCSP介绍

CP_nets 与近几年的研究热点——带有软约束满足问题 SCSP（Soft Constraint Satisfaction Problem）具有很强的关联性。下面给出与 SCSP 有关的受限半环及 SCSP 的定义，然后通过规约技术将 CP_nets 上的强占优测试转化为 SCSP 中解的优劣判断问题，从而增强部分 CP_nets 的表达能力。

定义 12. 受限半环（c-semiring）[2,3] $S = \langle A, +, \times, 0, 1\rangle$是一个包含两个运算+和×的代数结构，其中

（1）$\langle A, +, 0\rangle$是一个满足交换律和幂等律的独异点，且 0 是+的幺元，1 是+运算的零元。

（2）$\langle A, \times, 1\rangle$是一个满足交换律的独异点，1 是×运算的幺元，0 是×运算的零元，并且×运算对+运算满足分配律。

定义 13.

（1）SCSP是一个四元组的约束满足问题$P = \langle C, S, V, D \rangle$，其中S是定义 12 所示的受限半环，V是决策变量的集合，D是V的定义域，即$D = Dom(V)$。其中C 是所有形如$\langle Con, Util \rangle$的约束集合，即一个约束是二元组$\langle Con, Util \rangle$，$Con \subseteq V$是受限的决策变量集，$Util$是约束$Con$的效用函数（Utility Function），即$Util$：$D^{|Con|} \rightarrow A$。$Util$代表约束Con所产生的效用大小（约束强弱，偏好大小），它将一个约束的定义域Con的一个值映射为半环数据域A中的一个元素。

（2）给定一个软约束满足问题$P = \langle C, S, V, D \rangle$，$P$的一个解$Sol$是所有对V的赋值，解的效用值（偏好值）是$Sol$在所有限制上的效用值的乘积，即$Util(Sol) = \times_{c \in C} Util_c(Sol \downarrow_{Conc})$，其中$Sol \downarrow Conc$代表解$Sol$在约束定义区域$Conc$上的投影[2, 3]，$\times$是半环S中的$\times$运算。

例 6:几个常见的基于受限半环的软约束满足问题如下[3]。

（1）经典的约束满足问题（Constraint Satisfaction Problem）是基于半环$S_{CSP} = \langle \{false, true\}, \vee, \wedge, false, true \rangle$来求解。

（2）模糊约束满足问题（Fuzzy Constraint Satisfaction Problem）是基于半环$S_{FCSP} = \langle [0,1], Max, Min, 0, 1 \rangle$来求解。

（3）带权的约束满足问题（Weight Constraint Satisfaction Problem）是基于半环$S_{WCSP} = \langle Z^+, Max, +, 0, +\infty \rangle$来求解。

（4）词典序的约束满足问题（Lexicographic Constraint Satisfaction Problem）是基于半环$S_{LOCSP} = \langle A, Max_s, Min_s, MAX, 0 \rangle$来求解。

例 7: 给出一个模糊约束满足问题 P，以理解上述一些定义。

$P = \langle C, S, V, D \rangle$，其中$S = \langle [0,1], Max, Min, 0, 1 \rangle$，$V = \{X, Y, Z\}$，$Dom(X) = Dom(Y) = Dom(Z) = \{a, b\}$，约束集$C = \{C_{XY}, C_{YZ}\}$，其中$C_{XY} = \{<aa,0.4>, <ab, 0.1>, <ba, 0.3>, <bb, 0.5>\}$，$C_{YZ} = \{<aa,0.4>, <ab, 0.3>, <ba, 0.1>, <bb, 0.5>\}$。解$XYZ = bbb$（即$X= b$，$Y= b$，$Z= b$）是最优解，因此有$bbb > abb$。

5.2 Soft Constraint与CP_nets的关系

CP_nets是一种定性偏好断言，它的一个缺点是强占优测试为指数级复杂度。根据Bistarelli [3]的论述知，SCSP的优点是可在多项式时间完成解的优劣判断。可见CP_nets的强占优测试和SCSP上的解的优劣判断具有互补性：CP_nets上的强占优测试的缺点恰好是SCSP上解的优劣判断的优点。倘若能将部分CP_nets上的强占优测试问题在多项式时间规约为SCSP上的解的优劣判断问题，则CP_nets上的定性关系比较就可转换为SCSP上的定量数值比较，这样就可解决部分CP_nets上强占优测试的指数级复杂度问题。下面给出从无环CP_nets的强占优测试规约向SCSP下的解的优劣判断问题规约的方法。

5.3 从CP_nets向Soft Constraint的归约

规约是一种重要的计算机技术，它将一个不熟悉的，难以求解的新的计算问题，转化为一个熟悉的，容易求解的另一个领域的问题。从 CP_nets 向 SCSP 规约就是在保留 CP_nets 核心特性的基础上，将 CP_nets 上的强占优测试转化为 SCSP 上效用值的定量比较。

受数据库中关系分解理论的启发[21]（关系分解要保持原有的函数依赖和实现无损连接）[1]，我们在转化过程中要遵循如下原则：（1）**Ceteris Paribus语义保持**：即在原CP_nets图中条件偏好表$CPT(X_i)$所定义的条件偏好无关在SCSP中的语义要保持。（2）**强占优保持**：即在原CP_nets图中具有强占优关系的两个配置在SCSP中仍然具有解的优劣关系。

5.3.1 求取 SCSP 中的约束

为了将CP_nets下的强占优测试转变为SCSP中解的优劣判断问题，首先要从CP_nets图$N = <V, CE>$中构造一个带有软约束的满足问题$P = \langle C, S, V, D \rangle$。由于$P$中的V就是N中的V，D是V的定义域，S是一个受限半环，本文取带权约束半环$S = S_{WCSP} = \langle Z^+, Max, +, 0, +\infty \rangle$，因此转换的核心是构造$P$中的约束集C。

由于一个约束$c = \langle Con, Util \rangle$包含两部分$Con$和$Util$（一个约束本质上是一个函数），其中$Con$是约束$c$的定义域，$Util$是约束$c$的值域，该值域是一个从$D^{|Con|}$ 到环中A上的函数。因此构造一个约束的核心是先构造每个约束的变量集$Con \subseteq V$，然后设置Con的效用值$Util(Con)$，下面给出通过CP_nets构造SCSP中所有约束C的规则。

规则 1. CP_nets图$N = <V, CE>$，若$X_i \in V$，则从属性X可构造一个约束Con_{Xi}，称Con_{Xi}为属性X_i所诱导的约束。约束Con_{Xi}的定义域为$Para(X_i) \cup \{X_i\}$，V中所有属性所诱导的约束的定义域为$\{ Para(X_i) \cup \{X_i\} \mid X_i \in V\}$。

从规则 1 知，N中每个属性和其父亲构成一个约束的定义域。约束的定义域很直观，因为Agent对属性X_i的各个取值的偏好取决于其父亲$Para(X_i)$的不同取值，因此$Para(X_i) \cup \{X_i\}$可作为一个约束的定义域。

对于图 2 的CP_nets来说，N中包含三个决策变量J，P，S。由于J和P没有父亲，其对应的约束的定义域为$\{J\}$和$\{P\}$，而S的父亲是J和P，其对应的约束的定义域为$\{J, P, S\}$。一个约束对其定义域的不同组合所形成的效用函数值（约束值，偏好值）也不一样。例如Agent对变量J 的不同取值的偏好不一样，$J_b > J_w$代表对黑颜色夹克的偏好大于对白颜色的夹克的偏好，可理解成效用函数值与定义域$\{J\}$有关；同样Agent对J，P和S的不同取值的偏好也不一样。$J_b \wedge P_b \wedge S_r > J_b \wedge P_b \wedge S_w$代表对黑夹克、蓝裤子配红衬衫的偏好大于对黑夹克、蓝裤子配白衬衫的偏好，我们可理解成偏好值与约束的定义域$\{J, P, S\}$有关。CP_nets表达的是定性偏好信息，当给出不同约束的效用值后，就可用定量观点考察不同约束的偏好值。

---

1 关系数据库中的表也可看作CP_nets，其中字段集为CP_nets的顶点集。表的完整性可看作是一种约束，但目前的关系数据库的完整性还没有出现CP_nets的条件偏好表所阐述的约束。因此基于CP-nets来扩充关系数据库中的约束，定会使关系数据库的理论得到发展。

规则 2. 考虑无环m值CP_nets图N = <V, CE >[1]，对N中所有顶点进行拓扑排序后保存在单链表L，则

（1）$X_i \in V$所诱导的约束的权值$Weig(Con_{X_i}) = m^{Index(X_i)}$，其中$Index(X_i)$为属性$X_i$在$L$中的位置（拓扑序列最后一个变量的位置为 0，倒数第二个变量的位置为 1，…，即类似于十进制数 958，8 的$index$值为 0，5 的$index$值为 1，9 的$index$值为 2（从右往左数，第 0 位，第 1 位，第 2 位，…）。

（2）若$CPT(X_i)$中一个偏好断言为“$v_1 \wedge v_2 \wedge \ldots \wedge v_k$：$x_{i1} > x_{i2} > x_{i3} > \ldots > x_{im}$”，其中“$v_1 \wedge v_2 \wedge \ldots \wedge v_k$”代表$< v_1, v_2, \ldots, v_k >$是$Para(X_i)$的一个取值（$X_i$有$k$个父亲），“$x_{i1} > x_{i2} > x_{i3} > \ldots > x_{im}$”代表对$x_{i1}$的偏好最大，…，对$x_{im}$ 的偏好最小。则约束Con_{X_i}的序列值（用$Sequ$来表示）为$Sequ(<v_1, v_2, \ldots, v_k, x_{i1}>) = m-1$，$Sequ(<v_1, v_2, \ldots, v_k, x_{i2}>) = m-2$，⋯，$Sequ(<v_1, v_2, \ldots, v_k, x_{ik}>) = 0$。可见约束$Con_{X_i}$的序列值$Sequ$最大为$m-1$（代表对$Dom(X_i)$中的一个最大偏好的取值），…，最小为 0（代表对最小偏好的取值）。

（3）若c是约束Con_{X_i}定义域内的一个取值，即$c \in Dom(Con_{X_i})$，且约束Con_{X_i}的权值为$Weig(Con_{X_i})$，序列值为$Sequ(c)$，则约束Con_{X_i}的效用函数值$Util(Con_{X_i}) = \{ Weig(Con_{X_i}) * Sequ(c) \mid c \in Dom(Con_{X_i}) \}$。

例 8: 求例 1 中 CP_nets 中各个变量所诱导的约束。

解：CP_nets图N中的约束的定义域就是具有父子关系的二元关系有序对（对于根结点来说，是一元关系），因此约束集$C = \{c_1, c_2, c_3\}$，$c_1 = <Con_1, Util_1>$，$c_2 = <Con_2, Util_2>$，$c_3 = <Con_3, Util_3>$。其中约束c_1的定义域$Con_1 = \{J\}$，约束c_1的权值$Weig_1 = Weig(Con_1) = 2^2 = 4$，约束$c_1$的序列值$Sequ_1 = Sequ(Con_1) = \{<J_b, 1>, <J_w, 0>\}$，约束$c_1$的效用值$Util_1 = Util(Con_1) = \{<J_b, 4>, <J_w, 0>\}$；约束$c_2$的定义域$Con_2 = \{P\}$，约束$c_2$的权值$Weig_2 = Weig_2(Con_2) = 2^1 = 2$，约束$c_2$的序列值$Sequ_2(Con_2) = \{<P_b, 1>, <P_w, 0>\}$，约束$c_2$的效用值$Util(Con_2) = \{<P_b, 2>, <P_w, 0>\}$，约束$c_3$的定义域$Con_3 = \{J, P, S\}$，约束$c_3$的权值$Weig(Con_3) = 2^0 = 1$，约束$c_3$的序列值$Sequ(Con_3) = \{<J_b P_b S_r, 1>, <J_w P_b S_w, 0>, <J_w P_b S_w, 1>, <J_w P_b S_r, 0>, <J_b P_w S_w, 1>, <J_b P_w S_r, 0>, <J_w P_w S_r, 1>, <J_w P_w S_w, 0>\}$。约束$c_3$的效用值$Util_3 = Util(Con_3) = Util(\{J, P, S\}) = Weig(Con_3) * Sequ(Con_3) = 1 * Sequ(Con_3) = \{<J_b P_b S_r, 1>, <J_w P_b S_w, 0>, <J_w P_b S_w, 1>, <J_w P_b S_r, 0>, <J_b P_w S_w, 1>, <J_b P_w S_r, 0>, <J_w P_w S_r, 1>, <J_w P_w S_w, 0>\}$。

5.3.2 在 SCSP 中进行解的优劣判断

将 CP_nets 规约为 SCSP 后，我们可将 CP_nets 上的强占优测试转化为 SCSP 上的解的优劣判断问题。两个 SCSP 中解的优劣判断类似于两个二进制数的大小比较。

例如要想比较二进制 11010011 与 10011011 的大小，首先将其转化为十进制，$11010011 = 1*2^7+1*2^6+0*2^5+1*2^4+0*2^3+0*2^2+1*2^1+1*2^0 = 211$，而 $10011011 = 1*2^7+0*2^6+0*2^5+1*2^4+1*2^3+0*2^2+1*2^1+1*2^0 = 155$。因为 $211 > 155$，所以 $11010011 > 10011011$。

同样要想在SCSP中判断解sol_1与sol_2的优劣，即需要判断$sol_1 > sol_2$成立吗。

规则 3. $P = \langle C, S, V, D \rangle$是一个软约束满足问题 SCSP，

（1）$S = \langle A, +, \times, 0, 1 \rangle$是一个约束半环，若对于两个不同的解$sol_1$，$sol_2 \in A$，有$sol_1 + sol_2 = sol_1$，称解$sol_1$优于解$sol_2$，即$sol_1 > sol_{12}$（>为+运算诱导的偏序关系）。

（2）当半环$S = S_{WCSP} = \langle Z^+, Max, +, 0, +\infty \rangle$，若对于两个不同的解$sol_1$和$sol_2 \in Z^+$，有$Max(sol_1, sol_2) = sol_1$，称解$sol_1$优于解$sol_2$，即$sol_1 > sol_2$（>为$Max$运算诱导的偏序关系）。

（3）当$P = \langle C, S, V, D \rangle$是通过 5.3.1 节的规约技术而得到的软约束满足问题，即$S = S_{WCSP} = \langle Z^+, Max, +, 0, +\infty \rangle$，C、V和D是通过规则 1 和规则 2 而得到的集合，则对于CP_nets中的两个不同的配置o_1和o_2来说，若$Max(Util(o_1), Util(o_2)) = Util(o_1)$，其中 $Util(o) = Util(o_{\downarrow Con X_1}) + Util(o_{\downarrow Con X_2}) + \ldots + Util(o_{\downarrow Con X_n})$。可形象把一个配置的效用值（偏好值）理解为配置中所有属性诱导的约束的效用值之和（半环S_{WCSP}中的+运算）。

例 9: CP_nets图N如例 1 所示，对配置o_2 =“$J_w P_w S_r$”和o_5 =“$J_b P_b S_w$”进行强占优测试，即问$N \vDash o_5 > o_2$成立吗？

解：例 1 的CP_nets图N的一个拓扑排序是“J，P，S”，结合例 8 可知，$Weig(Con_J) = 2^2$，$Weig(Con_P) = 2^1$，$Weig(Con_S) = 2^0$，序列值为$Sequ(o_{5\downarrow ConJ}) = 1$，$Sequ(o_{5\downarrow ConP}) = 1$，$Sequ(o_{5\downarrow ConS}) = 0$，而$Sequ(o_{2\downarrow ConJ}) = 0$，$Sequ(o_{2\downarrow ConP}) = 0$，$Sequ(o_{2\downarrow ConS}) = 1$，故$Util(o_5) = Util(o_{5\downarrow ConJ}) + Util(o_{5\downarrow ConP}) + Util(o_{5\downarrow ConS}) = Util(J_w) + Util(P_w) + Util(J_w P_w S_r) = 2^2*1+2^1*1+2^0*0 = 6$，而$Util(o_2) = Util(o_{2\downarrow ConJ}) + Util(o_{2\downarrow ConP}) + Util(o_{2\downarrow ConS}) = Util(J_b) + Util(P_b) + Util(J_b P_b S_w) = 2^2*0 + 2^1*0 + 2^0*1 = 1$，由于$Max(Util(o_5), Util(o_2)) = Max(6, 1) = 6 = Util(o_5)$，故$N \vDash o_5 > o_2$。

5.3.3 规约的特性

定义 14. 通过 5.3.1 节的规则 1 和 2 所实现的 CP_nets 向 SCSP 的转化，

（1）若N中的条件偏好表中有一个条件无关条目$y\,x_1\,z > y\,x_2\,z$，有对应的SCSP中解$y\,x_1\,z$优于解$y\,x_2\,z$，则称规约具有条件无关语义保持。

（2）若N中配置o_1 强占优于o_2，有对应的SCSP中的解o_1优于o_2，则称规约具有强占优保持。

定理 12. 通过 5.3.1 节的规则 1 和 2 所实现的规约满足条件无关语义保持和强占优关系保持。

证明：（1）先证条件无关语义保持，即证$y\,x_1\,z > y\,x_2\,z \Rightarrow Util(y\,x_1\,z) > Util(y\,x_2\,z)$。

1 本文探讨CP_nets中任意$X_i \in V$，都有$|Dom(X_i)| = m$，即每个决策变量有m个取值。对于变量X_i，$X_j \in V$，$|Dom(X_i)| \neq |Dom(X_j)|$的情况类似。

对于顶点X的条件偏好表CPT(X)来说，$y\,x_1\,z > y\,x_2\,z$是该表的一个条目，其中$y \in$Dom($Para$(X))，z$\in$Dom(V−{X, Y})，即z是V−{X,Y}中的任意一个取值。根据CP_nets向SCSP的转换规则 2 的（1），配置$y\,x_1\,z$和$y\,x_2\,z$的权值$Weig$一样，即$Weig(y\,x_1\,z) = Weig(y\,x_2\,z)$。根据转换规则 2 的（2），$y\,x_1\,z$的序列值大于$y\,x_2\,z$的序列值，即$Sequ\,(y\,x_1\,z) > Sequ\,(y\,x_2\,z)$，因此$Util\,(y\,x_1\,z) > Util\,(y\,x_2\,z)$。

（2）再证强占优关系保持，即证$o_1 > o_2 \Rightarrow Util\,(o_1) > Util\,(o_2)$。

设o_1与o_2第一个不同的属性为X，在拓扑排序得到的序列L中，排在X之前的属性集是V_1，则o_1与o_2在V_1上的值都相同，o_1与o_2在V_1所诱导的约束上的效用值也一样，因此只需证明o_1在V−V_1所诱导的所有约束的效用值之和大于o_2在V−V_1上所诱导的所有约束的效用值之和。由于$o_1 > o_2$，则o_1在X所诱导的约束上的序列值大于o_2在X所诱导的约束上的序列值，而o_1与o_2在属性X所诱导的约束上的权值$Weig$一样，根据规则 2 的（2），有o_1在X及其祖先上所诱导的约束上的效用值之和大于o_2在X及其祖先所诱导的约束上的效用值之和。根据$Weig$值的计算规则，即使o_2在V−{V_1, X}所诱导的约束的效用值之和大于o_1在V−{V_1,X}所诱导的约束的效用值之和，o_2在V−{V_1, X}上所诱导的约束的效用值之和也小于o_1在X上所诱导的约束的权值$Weig$，因此$Util(o_1) > Util(o_2)$。这正如二进制 1000 与 0111 的大小比较一样，前者高位的权值为 $1*2^3 = 8$，后者后三位的效用值之和为 7，都小于前者最高位的权值 8。

定理 13. 基于 CP_nets 向 SCSP 规约后的强占优测试是多项式的时间复杂度。

证明：显然在带权约束半环中进行解的优劣判断可在多项式时间内完成，而规约步骤中的拓扑排序也可在多项式时间完成，加之 CP_nets 中顶点的父亲个数有限，即求 N 中每个变量的约束也可在多项式时间内完成。这样在 CP_nets 中进行强占优测试的时间就等于转化的时间加上在 SCSP 中进行解的优劣判断的时间。而这两部分的时间都是多项式时间，因此定理成立。

5.4 相关工作的对比

CP_nets 已经成为一种重要的偏好表示工具，近 10 年世界许多学者都在研究与 CP_nets 有关的偏好处理问题。为了理解本文的特色和贡献，表 1 给出了本文与有关论文工作的对比，对比方面主要有：对 CP_nets 的表达能力分析了吗？理论上解决了任意 CP_nets 的强占优测试问题吗？解决和分析问题的方法简单直观吗？在实践中有一种可行的强占优测试方法吗？（即有一种多项式时间复杂度的近似方法吗）。

Table 1 Related work comparison for CP_nets

表 1 CP_nets 的相关工作对比

相关论文	主要工作	有 CP_nets 表达能力分析吗?	有 CP_nets 表达偏好的完备性的探讨吗？	理论上强占优测试全部解决还是部分解决？解决方法复杂吗？	强占优测试实际解决了吗？
Boutilier's works [1]	全面论述 CP_nets 的语法、语义和应用	没有	没有	少部分解决 复杂	没有
Goldsmith's works [22]	全面论述 CP_nets 强占优测试的复杂度求取	没有	没有	大全部解决 极其复杂	没有
Gelain's works [2]	论述了 SCSP 在缺失偏好下的处理方法	间接给出了 SCSP 的部分求解能力	没有	没有	没有
Bistarelli's works[3]	全面论述基于半环的约束满足问题的求解	没有	没有	没有	没有
Wilson's works[23]	提出了对 CP_nets 的扩展	仅有部分论述	没有	没有	没有
本文工作	**论述 CP_nets 的数学基础（结构、性质等）强占优测试的求取、规约及表达能力的论述**	**较全面**	**有**	**全部解决 简单**	**部分解决**

从表 1 可知，本文的主要优点在于首次比较全面地论述了 CP_nets 的表达能力，第一次用很一般的初等方法（改进的求传递闭包的 Warshall 算法）彻底解决了 CP_nets 的强占优测试问题，更重要的是，本文不仅给出了强占优测试的理论解，还给出了一种简单可行的实际解（向 SCSP 规约），它可认为是强占优测试的部分解或近似解，这间接提高了部分 CP_nets 的表达能力。

6 结论和未来工作

本文研究了Agent的偏好表示工具——CP_nets及其表达能力，通过对CP_nets导出图性质的研究，指出二值CP_nets上的$n*2^{n-1}$对可交换配置上的强占优测试可在线性时间内完成，但对于不是可交换的配置进行强占优测试时，其复杂度为$O(4^n)$，第一次以初等方法给出CP_nets的强占优测试的算法及其复杂度。为了解决强占优查询的指数复杂度问题，本文将其转化为受限半环上的比较操作，从而将CP_nets上的指数复杂度的定性比较转化为约束半环上的多项式复杂度的定量比较，间接增强了部分CP_nets的表达能力。更为重要的是，本文较为全面地论述了CP_nets的表达能力，指出CP_nets适合不完全信息下的定性偏好推理，虽然有些CP_nets并不能表达出所有的强占优关系，但借助于与Agent的交互，可将CP_nets没有表达出的强占优关系通过交互计算而得到。

进一步研究方向有三：（1）研究CP_nets图的其他数学性质，如可满足性，一致性等，特别是深入研究CP_nets的完备性，重点研究CP_nets结构与其表达的强占优关系的联系，以进一步明确CP_nets表达能力严重依赖于N的图结构；（2）基于CP_nets

的表达能力，设计一个在条件偏好情况下的CP_nets求解器，从客户的偏好断言中求出其表达的含义，从而投其所好，为设计个性化的产品推荐系统原型奠定基础。（3）结合文献[23]提出的偏好表示方法，研究如何进一步增强CP_nets的表达与推理能力，并将CP_nets的表达能力与Game Theory的表达能力进行对比。

致谢 在此，我们向对本文的工作给予支持和建议的同行，尤其是各位审稿专家对本文提出的证明思路，问题描述等意见表示感谢.

References:

[1] Boutilier, C, Brafman, R, Domshlak, C, Hoos, H, Poole, D. CP-nets: a tool for representing and reasoning with conditional ceteris paribus statements. Journal of Artificial Intelligence Research, 2004, 21: 135–191.

[2] Gelain M, Pini M S, Rossi F, Venable K B, Walsh T, Elicitation strategies for soft constraint problems with missing preferences: properties, algorithms and experimental studies, Artificial Intelligence, 2010, 174(5-6) :270–294.

[3] Bistarelli S, Montanari U, and Rossi F. Semiring-based constraint solving and optimization. Journal of the ACM, 1997, 44(2): 201–236.

[4] Koriche F, Zanuttini B. Learning conditional preference networks with queries. In Proc. 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), 2009, 1930–1935.

[5] Lang J, Mengin J. The complexity of learning ceteris paribus separable preferences. In Proc. 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), 2009.848–853

[6] Conitzer, V. Eliciting single-peaked preferences using comparison queries. Journal of Artificial Intelligence Research, 2009, 21: 135–191.

[7] Chevaleyre Y, Koriche F. Learning ordinal preferences on multi-attribute domains: the Case of CP-nets. To appear in the book Preference Learning [M] (J. Fürnkranz and E. Hüllermeier, eds.), Springer-Verlag. 2010.

[8] Tang Pingzhong, Lin Fangzhen. Computer-aided proofs of Arrow’s and other impossibility theorems. Artificial Intelligence, 2009, 173(8–9):1041–1053.

[9] Lang, J. Logical preference representation and combinatorial vote. Annals of Mathematics and Artificial Intelligence, 2004, 42(1): 37–71.

[10] Conitzer, V, Sandholm, T and Lang, J. When are elections with few candidates hard to manipulate, Journal of the ACM, 2007, 54(3): 1–33.

[11] Walsh, T. Uncertainty in preference elicitation and aggregation. In Proceedings of the National Conference on Artificial Intelligence (AAAI 2007), Vancouver, BC, Canada, 2007, page 3–8.

[12] Zuckerman M, Procaccia A D, Rosenschein J S. Algorithms for the coalitional manipulation problem. Artificial Intelligence, 2009, 173(8-9): 392–412.

[13] Liu Jinglei, Zhang Wei. An $O(2.983^n)$ time complexity algorithm for optimal coalition structure generation. Journal of software. 2011, 22 (in Chinese). 刘惊雷, 张伟. 一个$O(2.983^n)$时间复杂度的最优联盟结构生成算法.软件学报. 20011, 22.

[14] Domshlak, C, Prestwich, S, Rossi, F, Venable, K and Walsh, T. Hard and soft constraints for reasoning about qualitative conditional preferences. Journal of Heuristics, 2006, 12(4/5), 263–285.

[15] Zhang Zhizheng, Xing Hancheng. A preference logic based on various kinds of preferences. Journal of software. 2007, 18(11): 2728−2739 (in Chinese). 张志政, 邢汉承, 王蓁蓁, 倪庆剑. 一种基于多类型偏好的偏好逻辑.软件学报. 2007, 18(11): 2728−2739.

[16] Brafman, R, Domshlak, C, & Shimony, E. On graphical modeling of preference and importance. Journal of Artificial Intelligence Research, 2006, 25: 389–424.

[17] Rossi, F, Venable, K. B and Walsh, T. Mcp_nets: Representing and reasoning with preferences of multiple agents. In Proc. AAAI 2004. AAAI Press.

[18] Brafman R, Domshlak C, Preference handling -- an introductory tutorial. AI Magazine, 2009, 30(1): 58-86.

[19] 袁崇义, 屈婉玲译. 离散数学及其应用(第 4 版). 北京: 机械工业出版社.2001: 393–396.

[20] 傅育熙. 计算机科学的基础、结构和问题. 中国计算机学会通讯. 2010, 6(3) 44–46.

[21] 王珊. 数据库原理. 北京: 高等教育出版社. 2005: 393–396.

[22] Goldsmith, J, Lang, J, Truszczy´nski, M, Wilson, N. The computational complexity of dominance and consistency in CP-nets. Journal of Artificial Intelligence Research, 2008, 33:403–432.

[23] Wilson N. Extending CP-nets with stronger conditional preference statements. In Proc. AAAI 2004, 735−741.

[24] Boutilier, C, Brafman, R, Domshlak, C, Hoos, H, Poole, D. Preference-based constrained optimization with CP-nets. Computational Intelligence (Special Issue on Preferences in AI and CP), 2004, 20 (2), 137–157.