

领域信息提取的缩规划图算法

柴啸龙

CHAI Xiao-long

广东商学院 数学与计算科学学院, 广州 510320

Department of Mathematics and Computing Science, Guangdong University of Business Studies, Guangzhou 510320, China

E-mail: chaixiaolongok@163.com

CHAI Xiao-long. Shrink planning graph algorithm based on domain information mining. *Computer Engineering and Applications*, 2009, 45(1): 22-24.

Abstract: The traditional technique of graph planning has a bottle-neck of efficiency when the algorithm dealing with the large scale problem. The reason is that almost all intelligence planning problems are NP hard, and the calculating quantity is a blow to the algorithm by degrees. Some improvements of the graph planning techniques are made in this paper as follows. First, the domain information mining is used in the algorithm. Second, the shrink planning graph also with the corresponding algorithm are presented based on the heuristic function from domain information. The algorithm can rank the different expanding branch, and prune some expanding branch with little hope. A related experiment of planning shows that the tactic is work.

Key words: intelligence planning; planning graph; shrink planning graph; heuristic information; domain knowledge

摘要: 传统的图规划技术在处理规模较大的智能规划问题时, 由于计算量的递增爆炸, 导致算法在规划问题上容易出现效率瓶颈。对图规划技术进行了一些改进: (1) 加入领域信息的动态提取和使用; (2) 提出了缩规划图的概念和算法, 通过引入基于领域信息的启发式函数, 对不同的扩展分支进行优选排序, 剪除一些执行希望小甚至不合理的扩展分支, 从而提高了系统执行效率。实验表明该策略是有效的。

关键词: 智能规划; 规划图; 缩规划图; 启发式信息; 领域信息

DOI: 10.3778/j.issn.1002-8331.2009.01.006 文章编号: 1002-8331(2009)01-0022-03 文献标识码: A 中图分类号: TP18

1 前言

智能规划是人工智能中的重要研究领域, 而图规划技术则是近年来在智能规划领域中出现的一种比较高效的规划算法^[1], 传统的图规划技术是一种完全扩展规划图的方式, 在处理的问题规模较大时, 由于递增爆炸的计算量, 导致算法在计算规模较大的规划问题时容易出现效率瓶颈, 在路径搜索等智能规划问题中往往也会遇到相似的问题^[2-7]。本文试图在以下方面对图规划技术进行一些改进: (1) 对领域信息的动态提取和使用; (2) 引入基于领域信息的启发式函数, 对不同的扩展分支进行优选排序, 剪除一些执行希望小甚至不合理的一些扩展分支, 从而提高系统的执行效率。

2 基本概念

定义 1 (规划问题)^[1] 规划问题 P 是四元组 (s_0, g, A, Σ) , 其中 s_0 为系统当前状态, 由一些命题进行描述, g 为需要实现的目标命题集, A 为动作集, Σ 包括动作集与状态命题集间的前件映射, 后件映射, 状态命题之间和动作之间的互斥约束关

系。 $\forall a \in A, Precond(a)$ 称为是动作 a 的前提条件, 它是一个命题集, 只有 $Precond(a)$ 中所有命题同时成立时, a 才可以执行。 $Effect(a)$ 称为 a 的动作效果, 它会否定或删除一些旧的命题条件, 同时产生一些新的命题条件。规划解是一个动作集的序列, 可以实现系统由初始状态逐步实现目标状态。

定义 2 (动作的互斥)^[8] 互斥的动作不能同时被执行。称动作 a_1 和 a_2 是互斥的, 若满足下列条件之一:

- (1) $\exists p_1 \in Effect(a_1), \exists p_2 \in Effect(a_2)$, 使得 p_1 和 p_2 是相互否定的命题公式。
- (2) $\exists p_1 \in Effect(a_1), \exists p_2 \in Precond(a_2)$, 使得 p_1 和 p_2 是相互否定的命题公式。
- (3) $\exists p_1 \in Precond(a_1), \exists p_2 \in Effect(a_2)$, 使得 p_1 和 p_2 是相互否定的命题公式。
- (4) $\exists p_1 \in Precond(a_1), \exists p_2 \in Precond(a_2)$, 使得 p_1 和 p_2 是相互否定的命题公式。
- (5) $\exists p_1 \in Precond(a_1), \exists p_2 \in Precond(a_2)$, 且 p_1 和 p_2 分别是由互斥的动作 b_1 和 b_2 产生。

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60773201); 广东省自然科学基金(the Natural Science Foundation of Guangdong Province of China under Grant No.06301003)。

作者简介: 柴啸龙(1980-), 男, 博士生, 讲师, 研究领域为智能规划, 知识工程与应用。

收稿日期: 2008-07-09 **修回日期:** 2008-08-18

定义 3(完全规划图) 规划问题 $P=(s_0, g, A, \Sigma)$ 。构建一个由命题层和动作层交替出现的规划图如下: 令 P_0 为描述 s_0 的所有命题。称为第 0 层命题层, 如果第 i 层命题层 P_i 已经得到, 则规定第 $i+1$ 层命题层 P_{i+1} 和第 $i+1$ 层动作层 A_{i+1} 如下: 如果 $\exists a \in A, Precond(a) \subseteq P_i$, 则令 $Effect(a) \subseteq P_{i+1}$ 。同时令: $a \in A_{i+1}$ 。这样得到的规划图称为完全规划图。

定义 4(缩规划图) 设 G 是规划问题 $P=(s_0, g, A, \Sigma)$ 所产生的完全规划图。把 G' 称为 G 的缩规划图, 如果它满足下列条件: (1) G' 是 G 的真子图, G' 中所有的命题层和动作层都是 G 中相应命题层和动作层的子集; (2) G' 的第 0 层命题层和 G 的第 0 层命题层完全相同; (3) G' 的最后一层命题层中包含了所有的目标命题; (4) 如果 A_{i+1}' 是 G' 的第 $i+1$ 层命题层, 则以下

两式成立: $\bigcup_{a \in A_{i+1}'} Precond(a) \subseteq P_i'$, $\bigcup_{a \in A_{i+1}'} Effect(a) \subseteq P_{i+1}'$ 。其中 P_i' 和 P_{i+1}' 分别是 G' 中第 i 层和第 $i+1$ 层命题层。

定义 5(衍生命题) 在任意状态下, 命题 P 成立的必要命题条件称为 P 的衍生命题。例如 $Alive(Bird)$ 是 $Flying(Bird)$ 的衍生命题。

衍生命题可以通过规则或基于领域信息来确定, 通过衍生命题的使用, 可以提高使用命题的灵活性。

定义 6(触发命题) 动作 a 的前提条件中只剩命题 P 未得到满足, 或者动作 a 的所有前提条件都已得到满足, 虽然命题 P 的成立与否都不会妨碍动作 a 的执行, 但是基于领域信息, 最好等命题 P 成立过以后再执行 a , 这样的命题 P 称为动作 a 的触发命题。动作 a 如果有多个触发命题, 把 a 所有的触发命题称为触发集。

动作的触发命题和动作的前提条件命题是不同的。由触发命题构成的触发信息, 可以由领域信息转化得到, 另外在系统执行中, 通过归纳方式发现的一些中间必然状态或必然成立的命题, 也可以作为相关动作的触发命题, 以此来实现系统通过实际运行, 自动化地挖掘领域信息, 从而改进系统的执行效率。

3 基于领域信息提取的启发式缩规划图规划系统

系统的设计主要分六部分, 下面将系统的主要设计部分按步骤模块的方式叙述如下:

步骤 1 初始化, 将所要解决的实际问题转化为一个规划求解问题 (s_0, g, A, Σ) 。建立系统的初始状态, 目标状态, 动作模型, 其中动作模型包括动作集合、动作的定义、动作和状态的转化映射关系, 以及互斥信息等。

步骤 2 将已有的领域信息描述转化为约束谓词公式, 或者互斥规则。

步骤 3 引入一个启发式评估函数 $H(s, a)$, 功能是对不同状态 s 下可以有效执行的不同动作 a 给出量化评估。好的评估函数不仅可以筛选出每一个状态下可以执行的动作, 并对动作的执行进行优选排序, 从而可以提高系统的效率。而且有时还可以避免一些无限计算问题, 如例 1 所示。

例 1 图 1 是 BlocksWorld 领域中著名的 Sussman 问题。

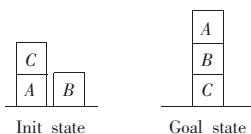


图 1 Sussman 问题

如果使用传统的规划算法, 而且不引入任何策略的话, 有可能会进入“死循环”的无限计算中。考虑引入下面的一个启发式函数 $H(a, s)$, 定义如下:

定义 7 函数 $H(a, s)$ 是一个对当前状态 s 下执行动作 a 的合适性的量化评估, a 所取得的函数值越小, 表明在状态 s 下越适合选择动作 a 。

- (1) 令 $H(a, s)=0$, 若满足 $on(C, table) \in Effect(a, s)$;
- (2) 令 $H(a, s)=1$, 若满足 $on(B, C) \in Effect(a, s)$;
- (3) 令 $H(a, s)=2$, 若满足 $on(A, B) \in Effect(a, s)$;
- (4) 其余情况, 令 $H(a, s)=3$ 。

每一状态下, 在所有的可以使用的动作中, 选择使 $H(a, s)$ 取值最小的动作 a 作为在当前状态 s 下执行的动作。易验证, 这种方法不仅顺利解决了 Sussman 问题, 而且所用的动作步骤是最短的。

步骤 4 根据动作模型进行规划图的扩展, 其中在进行扩展的时候, 根据启发式评估函数不断地通过优选只进行部分扩展, 生成一个最后一层命题层包含所有目标公式的缩规划图。启发式评估函数主要是对同位互斥动作之间作出优选排序, 下面给出同位互斥动作的定义:

定义 8(同位互斥动作) 设动作集 $A=\{a_1, \dots, a_k\}$ 。如果 $\forall a_i, a_j \in A$, 都有: $Precond(a_i)=Precond(a_j)$, 同时 $\exists p_1 \in Precond(a_i), \exists p_2 \in Precond(a_j)$, 使得 p_1 和 p_2 是互斥的命题对。则称动作集 A 中的任意两个动作互为同位互斥动作, 称 A 为同位互斥动作集。

系统在进行规划图的扩展时, 对出现同位互斥动作集发生扩展的情形, 如图 2 所示, 可通过基于领域信息所设定的启发式评估函数对各个同位互斥动作进行量化评估, 优选出最有希望的动作进行扩展。对希望小的动作不扩展或推迟扩展, 这样可大大提高系统的求解效率, 这些思想在传统规划图算法中是没有的。

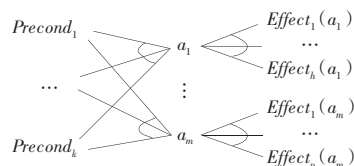


图 2 同位互斥动作在规划图中的扩展

步骤 5 领域信息的挖掘和提取

在系统的执行过程中, 如果发现有一些中间环节和状态是必须实现的, 即通过所有的扩展途径发现都必须取得的命题或状态, 可以把这一结果提取为一个领域信息, 并把它作为领域规则以谓词公式的形式进行描述, 然后把它加入到系统执行的约束公式或互斥规则中, 也可以把它作为相关动作的触发命题, 基于新的信息进行规划图的扩展调整, 把执行希望不大或不必要的动作和状态分支进行缩减或推迟扩展, 可以起到进一步减少后续命题层规模的效果, 从而提高后面基于规划图进行解提取部分的执行效率。

步骤 6 进行规划解的提取, 分两种情况: (1) 如果得到的是缩规划图, 进行规划解的提取, 并返回提取得到的规划解, 如果提取失败, 则放宽扩展限制, 并返回步骤 4; (2) 如果得到的是完全规划图, 进行规划解的提取。并返回提取得到的规划解, 如果提取失败, 则可以断定问题无解。

在规划图上进行规划解的提取时, 要求规划图中最后一层命题层 P_k 满足 $P_k \supseteq g$ 。提取的执行过程如下: 在最后一层动作

层 A_k 中选择尽量少的动作子集 π_k , 使:

- (1) $\forall a_1, a_2 \in \pi_k, a_1$ 与 a_2 在 A_k 中不是互斥的动作;
- (2) $\bigcup_{a \in \pi_k} Effect(a) \supseteq g$.

然后将 $\bigcup_{a \in \pi_k} Precond(a)$ 当成上面的 g , 继续向前层做倒序提取, 按同样方法选择 A_{k-1} 中动作子集 π_{k-1} , 若提取失败, 则回溯到上一步并重新选择动作子集, 直至提取出一个完整的动作集有序序列, 即为规划解。

4 实验

考虑如下规划问题, 在一个由 4×4 方格组成的迷宫图中, 三个物体 A, B, C 需要从它们的初始位置 As_0, Bs_0, Cs_0 分别移动到目标位置 Ag, Bg, Cg , 每步可移动一格, 也可选择静止不动。移动方向是上下左右四方向中的一个, 不能穿墙而过, 每一步中三物体可同时移动, 不能分先后顺序。求这一多主角迷宫问题的规划解。

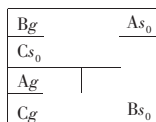


图3 多主角迷宫图的规划问题

步骤1 初始化。 首先将迷宫图中的 16 个方格位置数字化, 用 (i, j) 表示第 i 行第 j 列的方格。 $X(i, j)$ 表示物体 X 在方格 (i, j) 的位置上。所求解的规划问题 $P = ((A(1, 4) \wedge B(4, 4) \wedge C(2, 1)), (A(3, 1) \wedge B(1, 1) \wedge C(4, 1)), A, \Sigma)$, 其中动作集 $A = \{a_1, a_2, a_3, a_4, a_5\}$, 分别表示 MoveUp, MoveDown, MoveLeft, MoveRight 和 NoMove。

Σ 包含了动作和状态之间的映射关系。如在 $A(1, 4)$ 上执行动作 a_3 可以得到状态 $A(1, 3)$ 。执行该动作的前提条件是方格 $(1, 3)$ 上在动作 a_3 执行前和执行后都没有 A 以外的别的物体出现, 否则就是一种互斥。即 $A(1, 4)$ 和 $Empty(1, 3)$ 是动作 a_3 对 $A(1, 4)$ 执行的前提条件, 该动作的产生的效果条件则是 $NotEmpty(1, 3)$ 和 $A(1, 3)$, 同时该动作删除了条件 $Empty(1, 3)$ 。

Σ 也包含了对动作的约束关系, 如对 $B(4, 4)$ 不能执行动作 a_2 , 也包含了动作之间的互斥关系, 如同时对 $A(2, 3)$ 执行动作 a_4 和对 $B(3, 4)$ 执行动作 a_1 是一种互斥。因为它们会引起方格 $(2, 4)$ 上同时有两个物体的情形。

需要注意的是, 对于形如 $A(2, 3) \wedge B(2, 2) \wedge Empty(2, 4)$ 这样的状态情形, 系统认定对 $A(2, 3)$ 和 $B(2, 2)$ 两个物体同时执行动作 a_4 也是一种互斥情形。原因是 $A(2, 3)$ 会产生一个衍生条件 $NotEmpty(2, 3)$ 。而对 $B(2, 2)$ 执行动作 a_4 需要的一个前提条件是 $Empty(2, 3)$, 由于 $Empty(2, 3)$ 和 $NotEmpty(2, 3)$ 两个命题条件发生了互斥。因此对 $A(2, 3)$ 和 $B(2, 2)$ 两个物体不能同时执行动作 a_4 , 这和前面所叙述的多个物体可同时移动, 但不能分先后顺序是一致的。如果一定要对 $A(2, 3)$ 和 $B(2, 2)$ 都执行动作 a_4 , 只能把它们拆分在两步中分别进行。

步骤2 将已有的一些领域信息描述转化为约束谓词公式或者互斥规则。

不可以对 $X(i, j)$ 和 $Y(i, j+1)$ 或者 $X(i, j)$ 和 $Y(i, j-1)$ 同时

执行动作 a_3 , 也不可以同时执行动作 a_4 。不可以对 $X(i, j)$ 和 $Y(i-1, j)$ 或者 $X(i, j)$ 和 $Y(i+1, j)$ 同时执行动作 a_1 , 也不可以同时执行动作 a_2 。把这些领域信息描述为谓词公式并作为互斥规则存入 Σ 中。

步骤3 引入一个启发式评估函数 $H(X, s, a)$, 对不同状态 s 下在物体 X 上能有效执行的不同动作 a 给出量化评估。定义 $H(X, s, a)$ 为 X 在状态 s 下的方格位置和执行 a 后的方格位置的两个坐标分量差值的平方和。优选最小者, 三个物体 A, B, C 在不引起互斥的情形下可以分别用该函数进行计算。

步骤4 根据动作模型进行规划图的扩展, 进行扩展时, 根据启发式评估函数进行部分扩展。逐步生成缩规划图。启发式评估函数主要是对同位互斥动作进行。如命题 $B(2, 4)$ 若成立, 而且动作 a_2 和 a_3 的执行条件都已满足的话, 可以通过启发式评估函数 $H(B, s, a)$ 分别进行计算。 a_2 和 a_3 的函数值分别是 13 和 5, 显然优先对动作 a_3 进行扩展, 动作 a_2 可以不扩展或推迟扩展。

步骤5 领域信息的挖掘和提取

首先建立一个领域规则: 物体在进行动作扩展时, 要求不能回溯, 即使需要回溯, 其回溯后进一步的效果完全可以由别的扩展分支实现, 称这一规则为不可回溯规则。这一规则可以让很多分支停止扩展, 从而提高系统的执行效率。在系统的执行过程中, 先分别独立地对 A, B, C 进行扩展操作, 试图挖掘和提取出一些必然实现的命题作为领域信息。获得领域信息后, 再把三个物体合起来作为一个完整的实际问题进行实际扩展, 同时考虑更多的规则和领域约束。在这一规划问题中, 可先提取出领域信息: 命题 $B(3, 4)$ 必然实现。图 4 是得到这一领域信息所进行的动作扩展规划图的生成情况。

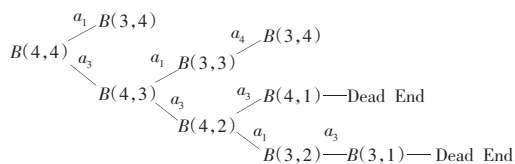


图4 领域信息“B(3,4)需要实现”的发现

同样的方式可以提取得到领域信息: 命题 $A(2, 4)$ 和 $C(2, 4)$ 都需要实现。由于命题 $B(3, 4)$ 和 $A(2, 4)$ 和 $C(2, 4)$ 都需要实现, 如果在某状态下发生了: $B(3, 4) \wedge A(2, 4)$ 或 $B(3, 4) \wedge C(2, 4)$, 由不可回溯规则可推理出问题无解。为避免这一情况, 需要使命题 $B(3, 4)$ 与 $A(2, 4)$, 以及 $B(3, 4)$ 与 $C(2, 4)$ 不出现同一命题层中。考虑到使 $A(2, 4)$ 或 $C(2, 4)$ 无法继续扩展的原因是 $B(3, 4)$ 的衍生命题 $NotEmpty(3, 4)$ 的成立。结合不可回溯规则和已经获得的领域信息, 可以对 B 一直执行动作 a_3 。因此 $A(2, 4)$ 的实现以及 $C(2, 4)$ 的实现可以成为对 $B(4, 4)$ 执行动作 a_1 的触发命题。利用获得的这一领域信息可以改进启发式函数 $H(X, s, a)$, 在命题 $A(2, 4)$ 和 $C(2, 4)$ 实现以前, A 和 C 的目标坐标应设为 $(2, 4)$, 实现以后新的目标坐标分别再设定回原定的目标值, B 的情况类似。

步骤6 通过获得的领域信息, 利用修订后的启发式函数通过动作扩展得到缩规划图, 并在缩规划图上进行规划解的提取, 如图 5 所示。限于篇幅, 只展示前面部分。其中因启发式函数数值较高而不扩展或至少延迟扩展的分支用 suspend 标记。第 0 层命题层中 $B(4, 4)$ 得不到 a_1 扩展分支的原因是其触发命题还没成立过。