软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(5):914−928 [doi: 10.3724/SP.J.1001.2011.03778]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

智能规划中的可纳子目标排序*

梁瑞仕[1,2+], 姜云飞[1], 边 芮[1,3], 吴向军[1,4]

[1](中山大学 信息科学与技术学院,广东 广州 510275)

[2](电子科技大学 中山学院 计算机学院,广东 中山 528402)

[3](广东商学院 公共管理学院,广东 广州 510320)

[4](中山大学 软件学院,广东 广州 510275)

Admissible Subgoal Ordering for Automated Planning

LIANG Rui-Shi[1,2+], JIANG Yun-Fei[1], BIAN Rui[1,3], WU Xiang-Jun[1,4]

[1](School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

[2](School of Computer, Zhongshan Institue, University of Electronic Science and Technology of China, Zhongshan 528402, China)

[3](School of Public Management, Guangdong University of Business Studies, Guangzhou 510320, China)

[4](School of Software, Sun Yat-Sen University, Guangzhou 510275, China)

+ Corresponding author: E-mail: liangruishi@foxmail.com

Liang RS, Jiang YF, Bian R, Wu XJ. Admissible subgoal ordering for automated planning. *Journal of Software*, 2011, 22(5):914−928. http://www.jos.org.cn/1000-9825/3778.htm

Abstract: This paper proposes an ordering relation named Admissible Subgoal Ordering (ASO). The definition of ASO is formalized, and its relative importance for incremental planning is discussed. Then, this paper introduces the notion of dependency relations over facts, and develops fact dependency graph technique that can approximate admissible ordering relations in polynomial time. Finally, an algorithm to compute subgoals sequence with admissible orderings is presented. All the ideas presented in the paper are implemented in the planning system ASOP, and the effectiveness of the techniques is demonstrated on the benchmarks of International Planning Competitions (IPC). The results show that these techniques can efficiently solve large planning problems and lead to a greater improvement in planning performance.

Key words: AI planning; subgoal ordering; admissible ordering relation; incremental planning; fact dependency graph technique

摘 要: 提出了一种称为可纳子目标排序(admissible subgoal ordering,简称 ASO)的排序关系,给出了可纳排序的形式化定义并讨论其对增量式规划的重要性.随后介绍了原子依赖关系理论和原子依赖图技术,能够在多项式时间内近似求解可纳子目标排序关系.最后给出了一种计算可纳子目标序列的算法.其所有思想已经在规划系统 ASOP 中实现.通过在国际规划大赛标准测试领域问题上的实验,其结果表明,该方法能够有效地求解大规模的规划问题,并能极大地改善规划性能.

关键词: 智能规划;子目标排序;可纳排序关系;增量式规划;原子依赖图技术

* 基金项目: 国家自然科学基金(60773201, 60970042); 电子科技大学中山学院博士科研启动基金(410YKQ03)
收稿时间: 2009-09-07; 定稿时间: 2009-11-04

中图法分类号: TP181 文献标识码: A

智能规划(AI planning)是人工智能的一个重要研究分支.它研究的主要问题模型是:如何在给定的动作模型(或者状态转换系统)框架下,自动求解出一个能从初始状态到达目标状态并且满足特定条件和约束的动作执行结构,例如串行规划、并行规划、条件规划、conformant 规划等.Bylander 等人[1]已经证明,对于经典的 STRIPS 规划问题,判定规划解的存在性和最优规划解的存在性的复杂度都是 PSPACE-complete.经过最近十几年来的发展,智能规划的研究已经取得了很大的进展,并成为当前人工智能的研究热点.

当前的高性能规划器,例如 FF[2],LPG[3],SGPLAN[4],FastDownward[5],Londex[6]等,由于规划方法不断改进,已经能够解决越来越多的规划问题.然而在很多大规模复杂的规划问题上,包括国际规划大赛(International Planning Competitions,简称 IPC)的标准测试领域问题,当前的规划器仍然面临着很大的挑战.提高规划系统效率的一个很重要的思想是,将大规模的原始规划问题进行分解,约简成多个更小、更简单的子问题,然后逐步求解.子目标排序就是体现该思想的一种有效方法.

规划问题的目标状态(原子集合)通常包含多个子目标原子,各个子目标之间一般都具有交互关系.如果这些子目标在实现(求解过程)上存在着先后顺序,则可以对它们进行排序,并划分到不同的子目标集合.这样,一个大的规划目标被分解成若干个具有先后顺序的子目标集合的序列,原始的规划问题也就被分解成多个子问题.在规划求解过程中,不必一次性求解所有子目标(子问题),只需按顺序逐步求解各个子目标,最终得到原始问题的规划解.例如,在 Block 领域中有 3 个积木块 A,B 和 C,该规划目标包括两个子目标原子:{(on A,B),(on B,C)}表示 A 在 B 上,B 在 C 上,那么在实现子目标(on A,B)之前,应该先实现(on B,C).于是,这两个子目标具有一种顺序实现关系(on B,C)→(on A,B).子目标排序问题就是研究子目标之间存在的顺序关系以及求解和应用这类顺序关系的方法.一个好的排序关系应该使每个子目标集合的求解尽量简单,而且能避免不必要的规划过程以及重复的规划.这样不仅能够提高规划效率,而且有助于提高优化规划解的质量.

已经有很多关于子目标排序的研究和算法,一些工作主要侧重于研究子目标排序的算法和应用,但对子目标之间的约束条件缺乏明确的形式化定义.Koehler 和 Hoffmann 等人在文献[7−9]中对子目标排序进行了深入研究,明确提出了几类排序关系(在本文第 4 节“相关工作”部分将详细加以介绍).然而,这些已有的排序关系对子目标之间的约束条件比较严格,仅仅代表了一类强约束的排序关系.从最近几届的国际规划大赛(IPC-3,IPC-4,IPC-5)的基准测试领域中可以发现,这种强约束的顺序关系只在很少的规划问题中存在.

本文提出了一种新的子目标排序关系,称为可纳排序(admissible subgoal ordering).可纳排序与已有的排序关系最大的不同在于,它弱化了子目标之间的约束条件,但它又是正确、有效的排序关系.可纳排序的一个很重要的优点是,它在大量规划领域问题中都存在,因此它具有普遍性.利用可纳排序关系能够有效地对规划问题进行划分和排序,从而降低问题的规模和复杂程度,有助于提高规划求解效率.

可纳排序问题是一类复杂的计算问题.本文介绍了原子依赖关系理论和原子依赖图技术,能够在多项式时间近似求解任意两个子目标原子之间的可纳排序关系.利用原子依赖图技术,本文给出了一种计算可纳子目标序列的思想及算法.可纳子目标序列能够被应用在基于目标议程(goal agenda)[7]的增量式规划算法中.

可纳排序的思想及其相关的技术已经在我们的规划器 ASOP(admissible subgoal ordering based planning system)中实现.我们选取国际规划大赛 IPC-3,IPC-4 和 IPC-5 的 4 个标准规划领域进行实验,并对实验结果进行了全面的对比和分析.结果表明,我们的方法能够有效地求解大规模的规划问题,并能显著地改善规划性能.

本文第 1 节介绍规划领域的定义和问题描述.第 2 节给出可纳排序的定义、计算可纳排序的方法以及如何使用可纳排序.第 3 节给出实验数据和对比.第 4 节讨论与本文相关的研究工作.最后给出本文的结论和未来的工作方向.

1 规划问题的定义和符号

本文采用 PDDL 语言来表示经典规划问题.一个规划问题 Q 可以表示为一个四元组:$Q=(F,A,I,G)$.其中,F 是

基原子事实(atomic facts)集合,A 是基动作(actions)集合,$I,G\subseteq F$ 分别为问题的初始状态和目标状态.以下简称基原子事实为原子,简称基动作为动作.每个动作 a 都由一个三元组$\langle pre(a),add(a),del(a)\rangle$构成,分别表示动作的前提表、增加表和删除表,它们都是原子的有限集合.在本文中,我们也将出现在增加表中的原子称为正效果(原子),把删除表中的原子称为负效果(原子).注意,根据经典规划的约定,动作中的负效果原子必须也是该动作前提表的一个原子.

状态是原子的集合.我们称原子 f 在状态 S 下成立(为真),当且仅当 $f\in S$.如果动作 a 的前提表在状态 S 下成立,即 $pre(a)\subseteq S$,则称 a 可应用于状态 S,且执行动作 a 的结果为 $exec(S,a)=S\cup add(a)\backslash del(a)$;否则,$a$ 不可应用于状态 S,即在 S 下不可执行.对于一个动作数目为 N 的动作序列 $p=\langle a_1,\ldots,a_N\rangle$,如果 $pre(a_1)\subseteq S$ 且 $S_1=exec(S,a_1)$,$pre(a_2)\subseteq S_1$ 且 $S_2=exec(S_1,a_2)$,…,$pre(a_N)\subseteq S_{N-1}$ 且 $S_N=exec(S_{N-1},a_N)$,则称 p 是状态 S 下一个可应用的规划(applicable plan),且规划长度为 N,执行该规划的结果为 $exec(S,\langle a_1,a_2,\ldots,a_N\rangle)=exec(exec(S,a_1),\langle a_2,\ldots,a_N\rangle)$.规划 p 中全部动作的集合记为 $A(p)=\{a_i|a_i \text{ is an action in } p,1\leqslant i\leqslant N\}$.

对于一个规划问题 $Q=(F,A,I,G)$以及一个规划 $p=\langle a_1,\ldots,a_N\rangle$,如果 $G\subseteq exec(I,\langle a_1,a_2,\ldots,a_N\rangle)$,则称$\langle a_1,a_2,\ldots,a_N\rangle$是问题 Q 的一个规划解(solution plan).并且,如果没有任何其他规划解的动作数目小于 N,则$\langle a_1,a_2,\ldots,a_N\rangle$是问题 Q 的一个最优规划解.一个状态 S 是可达的,当且仅当存在一个有效的规划π,能够从初始状态 I 到达 S,即 $S\subseteq exec(I,\pi)$.我们用记号 $S_{[e,\neg f]}$代表可达状态集合 $S_{[e,\neg f]}=\{s|e\in s\wedge f\notin s \text{ and } s \text{ is reachable from } I\}$.用 $P_{s\to[f]}$表示从可达状态 s 实现原子 f 的规划集合 $P_{s\to[f]}=\{p|f\in exec(s,p),s\in S_{[e,\neg f]}\}$,用 $OPT_{s\to[f]}$表示从可达状态 s 实现原子 f 的最优规划集合 $OPT_{s\to[f]}=\{p|p\in P_{s\to[f]} \text{ and } p \text{ is an optimal plan in } P_{s\to[f]}\}$.

2 可纳子目标排序

本节首先形式定义可纳排序关系,随后介绍计算可纳排序关系和可纳子目标序列的方法,最后给出一种结合可纳子目标排序的增量式规划算法.

2.1 可纳排序的定义

在形式化定义和讨论可纳排序关系之前,首先考察 ZenoTravel 领域的一个例子.初始状态和目标状态分别如图 1 所示.两个子目标原子及含义分别为:g_1:(at P1 City2),人 P1 在目标地点 City2;g_2:(at A1 City3),飞机 A1 在目标地点 City3.

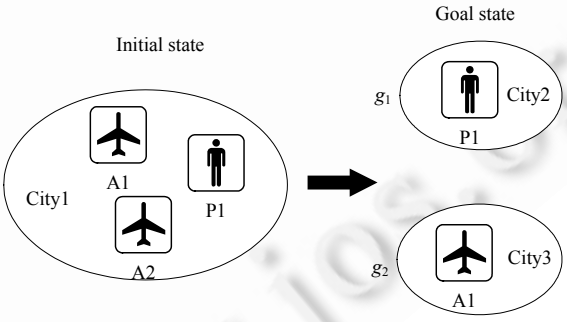


Fig.1 An example of ZenoTravel domain

图 1 ZenoTravel 领域的一个例子

在这个规划问题中,g_1 和 g_2 都是可以实现的(存在规划解).从子目标实现的顺序来看,既可以先实现 g_1 之后再实现 g_2,也可以先实现 g_2 之后再实现 g_1.也就是说,g_1 和 g_2 之间并没有强制的先后实现顺序,任何一种实现顺序都存在规划解.但是仔细观察发现,g_1 先于 g_2 实现((at P1 City2)→(at A1 City3))是一个相对合理的排序方式.这是由于人(P1)必须依赖飞机(A1 或者 A2)运送才能到达目标地点(任何运送的动作都需要以飞机的状态为前提),但飞机的飞行完全不依赖于人所处的位置(任何飞行的动作都不需要以人的状态作为前提).因此,这样的排

序能够确保 g_1 在实现之后,无须删除 g_1 也可以实现 g_2;并且在不删除 g_1 的情况下,总是可以找到比删除 g_1 而得到的规划解更好的解(因为删除 g_1 对实现 g_2 没有意义).

现在用形式化的规划语言来分析该排序问题.假设在初始状态 I 存在可应用规划 p_1 能够实现 g_1,并得到状态 $s_{g_1}=exec(I,p_1)$.又假设 p_2 是状态 s_{g_1} 下可应用的规划,能够实现子目标 g_2,并得到状态 $s_{g_2}=exec(s_{g_1},p_2)$.由于飞机的飞行完全不依赖于人所处的状态(地点),因此对于规划 p_2 中的任何动作,删除 g_1 对实现 g_2 没有任何帮助.也就是说,如果 p_2 中存在删除子目标 g_1 的动作,则完全可以从 p_2 中移除这类动作,仍然可以实现 g_2.换言之,如果 p_2 是一个最优规划动作序列,那么它肯定不包含删除 g_1 的动作.显然,这样能够确保已经先实现的子目标 g_1 不会被删除,即 $g_1\in s_{g_2}$.于是,两个子目标原子都能够实现.

上述例子中,(at P1 City2)和(at A1 City3)之间的顺序关系称为可纳排序关系.下面给出可纳排序的形式化定义.

定义 1(可纳排序). 已知规划问题 $Q=(F,A,I,G)$.令 $e,f\in G$ 分别是目标集合中的两个原子,e 和 f 之间具有可纳排序关系(admissible ordering relation),记为 $e<_{AO}f$,当且仅当以下条件同时成立:

(1) $S_{[e,\neg f]}\neq\varnothing\wedge\forall s\in S_{[e,\neg f]},P_{s\to[f]}\neq\varnothing$;

(2) $\forall s\in S_{[e,\neg f]},\forall p\in OPT_{s\to[f]},\neg\exists a\in A(p),e\in del(a)$.

其中,$S_{[e,\neg f]}$是可达状态集合$\{s|e\in s\wedge f\notin s$ and s is reachable from $I\}$;$P_{s\to[f]}$表示从可达状态 s 实现原子 f 的规划集合$\{p|f\in exec(s,p),s\in S_{[e,\neg f]}\}$;$OPT_{s\to[f]}$是从可达状态 s 实现原子 f 的最优规划集合$\{p|p\in P_{s\to[f]}$ and p is an optimal plan in $P_{s\to[f]}\}$,称为局部最优规划;$A(p)$是指规划 p 中所包含的动作的集合.

根据可纳排序,如果 $e<_{AO}f$,在规划求解过程中就可以将 e 和 f 划分在两个不同的规划步骤求解,先实现 e 后再实现 f.由定义的两个条件可知,可纳排序能够保证原来可解的子目标经过排序后仍然可解;并且在不删除已经实现的子目标的情况下,后面的子目标仍然能够实现.可纳排序要求在局部最优规划中不删除已经实现的子目标,这意味着在每个后续子目标的规划步骤中都可以找到局部最优解.

从第 1 个条件来看,如果 $S_{[e,\neg f]}$为空集,或者 $P_{s\to[f]}$为空集,则可纳排序 $e<_{AO}f$ 不存在.这是显然的.当 $S_{[e,\neg f]}$为空集时,意味着不存在 e 实现而 f 未实现的这类可达状态.因此,f 至少要与 e 同时实现或者在 e 之前实现;当 $P_{s\to[f]}$为空集时,意味着在某些或者任意可达状态 $s_{[e,\neg f]}$下,子目标 f 是无解的(不可达).也就是说,先实现 e 再实现 f 的排序方式将导致该规划问题不可解,这同样说明 e 在 f 之前实现是不恰当的.

可纳子目标排序关系提供了一种新的领域无关的子目标划分和排序的理论基础.与 Koehler 等人开发的具有强约束关系的排序理论相比,可纳排序弱化了两个子目标之间的约束关系,它代表了一类普遍存在的排序关系.大量规划问题(子目标之间)都存在可纳排序关系,这为利用可纳排序求解复杂的规划问题提供了广泛的应用基础.然而,自动提取规划问题中的可纳排序关系不是容易的事情.

计算任意两个子目标原子之间的可纳排序关系是一类困难而又复杂的问题,这是因为,从第 1 个条件来看,判定可达状态存在性和规划解存在性的复杂度都等价于判定规划问题的规划解存在性(PLANSAT 问题)的复杂度,具有 PSPACE-complete 复杂度;从第 2 个条件来看,判定最优规划不包含某类动作的问题,可多项式归约为判定规划问题是否具有最优规划解(PLANMIN 问题),同样具有 PSPACE-complete 复杂度.关于 PLANSAT 和 PLANMIN 问题复杂度的详细证明过程可以参考文献[1].

在本文中,我们重点关注计算可纳排序第 2 个条件的方法,而总是假定可纳排序的第 1 个条件成立.这主要是因为绝大部分的规划问题,包括 IPC 提供的基准规划问题以及现实生活中的规划问题,几乎都满足可纳排序的第 1 个条件,具有普遍性.

2.2 计算可纳排序

本节给出一种基于原子依赖关系理论的方法来计算可纳排序.首先介绍原子之间存在的依赖关系与非依赖关系.随后讨论最优规划理论的一些性质及其与依赖关系和非依赖关系的联系,并由此证明一个基于非依赖关系的可纳排序成立的充分条件.基于该充分条件,我们开发了原子依赖图技术,能够在多项式时间判定任意两个子目标原子之间的可纳排序关系.最后给出计算可纳子目标序列的思想和算法.

在下文的叙述中,为简化书写,除非特别声明,我们规定所讨论的规划问题为 $Q=(F,A,I,G)$,F 是基原子事实集合,A 是基动作集合,$I,G\subseteq F$ 分别为问题的初始状态和目标状态.所有的集合均不为空.并且,我们总是假设所讨论的规划问题满足可纳排序定义的第 1 个条件成立.这里不再赘述.

2.2.1 原子之间的依赖关系

定义 2(直接依赖关系). 已知原子 $u,v\in F$,动作 $a\in A$.如果 u 是 a 的一个正效果原子,v 是 a 的一个前提表原子,则 u 通过 a 直接依赖于 v,简称 u 直接依赖于 v.

我们用符号 $\rightarrow_a$ 来表示直接依赖关系,则 $u\rightarrow_a v$ 表示 u 通过 a 直接依赖于 v.在不引起歧义的情况下,我们简记为符号 $u\rightarrow v$.

直接依赖关系展示了动作的正效果原子和前提表原子之间存在的一种内在联系.显然,根据上述定义可知,对于任何一个动作,它的每一个正效果原子都将会依赖于该动作的每一个前提表原子.

命题 1. $\forall a\in A,\forall u,v\in F,(u\in add(a))\wedge(v\in pre(a))\Rightarrow(u\rightarrow_a v)$.

有了原子直接依赖关系,下面将递归定义原子依赖关系.

定义 3(依赖关系). 已知原子 $u,v\in F$,u 依赖于 v,当且仅当以下两个条件之一成立或者全部成立:

(1) u 直接依赖于 v;

(2) 存在一个原子 $w\in F$,u 依赖于 w,且 w 依赖于 v.

我们用符号⇢来表示原子依赖关系,则 $u\dashrightarrow v$ 表示 u 依赖于 v.对于定义 3 中的第 2 种情况,我们称为间接依赖关系.从递归定义不难发现,直接依赖关系可以看成是最基本的依赖关系,而间接依赖实质是由直接依赖关系不断组合而得到的.任何依赖关系(以及间接依赖关系)都可以分解为一系列直接依赖关系的组合.

给定规划问题 $Q=(F,A,I,G)$,它的依赖关系集合记为 $\Omega=\{\langle u,v\rangle|u\dashrightarrow v,u,v\in F\}$.显然,$\Omega$ 是建立在原子集合 F 之上的二元关系集合.不难发现,依赖关系具有传递性.

命题 2(依赖传递性). $\forall u,v,w\in F,(u\dashrightarrow w)\wedge(w\dashrightarrow v)\Rightarrow(u\dashrightarrow v)$.

由定义 3 的第 2 个条件可知,命题 2 显然成立.

对于任意两个原子 u 和 v,它们之间既可以通过某个动作具有直接依赖关系,也可以有间接依赖关系,或者二者兼而有之.依赖关系天然具有传递性,但不一定具有自反性和对称性,这与具体的问题描述有关.相反地,如果 u 不依赖于 v,则它们具有非依赖关系.

定义 4(非依赖关系). 已知规划问题 $Q=(F,A,I,G)$及其依赖关系集合 Ω.原子 $u,v\in F$,u 非依赖于 v,当且仅当 $\langle u,v\rangle\notin\Omega$.

我们用符号⇸表示非依赖关系,则 $u\nrightarrow v$ 表示 u 非依赖于 v.

现在我们来综合考察原子 u 和 v 之间依赖关系(及非依赖关系)的组合情况(见表 1):(1) u 依赖于 v,且 v 依赖于 u,则称 u 和 v 相互依赖;(2) u 非依赖于 v,且 v 非依赖于 u,即 u 和 v 之间不存在任何依赖关系,则称其为相互独立关系;(3) u 依赖于 v,但 v 不依赖于 u;或者相反,此时称为单向依赖关系.

Table 1 Combinations of dependency relations between two facts u and v

表 1 两个原子 u,v 之间的依赖关系组合情况

Dependency relation	$u\dashrightarrow v$	$u\nrightarrow v$
$v\dashrightarrow u$	u and v are interdependent	v unilaterally depends on u
$v\nrightarrow u$	u unilaterally depends on v	u and v are mutually independent

2.2.2 可纳排序关系的一个充分条件

本节讨论最优规划的相关性质以及与原子依赖关系的联系,由此推导出可纳排序关系的一个充分条件.

给定规划问题 $Q=(F,A,I,G)$,令 $e,f\in G$ 分别是两个子目标原子.假设在初始状态 I 存在一个规划 $p_{I\rightarrow[e]}$,能够实现 e,但没有实现 f,并得到可达状态 $s_{[e,\neg f]}=exec(I,p_{I\rightarrow[e]})$.并假设 $opt_{[e,\neg f]\rightarrow[f]}$ 是状态 $s_{[e,\neg f]}$ 下的一个最优规划,能够实现子目标 f,即 $f\in exec(s_{[e,\neg f]},opt_{[e,\neg f]\rightarrow[f]})$.令最优规划 $opt_{[e,\neg f]\rightarrow[f]}=\langle a_1,\ldots,a_N\rangle,a_i\in A,1\leqslant i\leqslant N$.为简化书写,引入如下记号: $P_i=pre(a_i),E_i=add(a_i),1\leqslant i\leqslant N$.

引理 1. $\forall i.1\leqslant i<N, f\notin E_i\wedge f\in E_N$.

证明:假设存在 $E_k(1\leqslant k<N)$满足 $f\in E_k$.令 $p'=\langle a_1,\ldots,a_k\rangle$,于是 $f\in exec(s_{[e,\neg f]},p')$.显然,由于 $k<N$,那么 p'的动作数目(k 个)比 $opt_{[e,\neg f]\to[f]}$要少,这与前面所述"$opt_{[e,\neg f]\to[f]}$是最优规划"矛盾.假设不成立.于是可得$\forall i,1\leqslant i<N, f\notin E_i$.又因为 $f\in exec(s_{[e,\neg f]},opt_{[e,\neg f]\to[f]})$,因此 $f\in E_N$.原命题得证. □

引理 2. $\forall i,\exists j,1\leqslant i<j\leqslant N, E_i\cap P_j\neq\varnothing$.

证明:假设存在 $E_k(1\leqslant k<N)$,对任意 $P_l(l=k+1,\ldots,N)$满足 $E_k\cap P_l=\varnothing$.这意味着($opt_{[e,\neg f]\to[f]}$中)动作 a_k 的增加表与动作 $a_{k+1},\ldots,a_N$ 的前提表的交集均为空.令 $p'=\langle a_1,\ldots,a_{k-1},a_{k+1},\ldots,a_N\rangle$(也就是从 $opt_{[e,\neg f]\to[f]}$中移去动作 a_k 而得).由于动作序列$\langle a_1,\ldots,a_{k-1}\rangle$是最优规划 $opt_{[e,\neg f]\to[f]}$的前面一部分,而$\langle a_{k+1},\ldots,a_N\rangle$中的每一个动作的前提表都与 a_k 的增加表无交集,因此 p'仍然是一个在状态 $s_{[e,\neg f]}$下可应用的规划(因为每个动作的前提表都能够得到满足).

而由引理 1 可知,$f\notin E_k\wedge f\in E_N$.于是可得 $f\in exec(s_{[e,\neg f]},p')$.显然,$p'$的动作数目($N-1$ 个)比 $opt_{[e,\neg f]\to[f]}$要少,这与"$opt_{[e,\neg f]\to[f]}$是最优规划"矛盾.假设不成立.于是得证. □

引理 3. $\forall i,1\leqslant i\leqslant N,\forall p\in P_i, f\dashrightarrow p$.

证明:

(1) 当 $i=N$ 时,由引理 1 可知 $f\in E_N$.由于 E_N 和 P_N 分别是动作 a_N 的增加表和前提表,它们的原子之间具有直接依赖关系,于是可得$\forall p\in P_N, f\dashrightarrow p$.

(2) 假设存在正整数 $k(k>1)$,满足$\forall i,k\leqslant i\leqslant N,\forall p\in P_i, f\dashrightarrow p$.

现在要证明:$\forall i,k-1\leqslant i\leqslant N,\forall p\in P_i, f\dashrightarrow p$.

由引理 2 可知:$\exists j,1\leqslant k-1<j\leqslant N, E_{k-1}\cap P_j\neq\varnothing$.令 $e\in E_{k-1}\cap P_j$.

因 $k\leqslant j\leqslant N$ 且 $e\in P_j$,由假设可知 $f\dashrightarrow e$.

又因为 $e\in E_{k-1}$,它将直接依赖于 a_{k-1} 的每一个前提表原子,即$\forall p\in P_{k-1}, e\dashrightarrow p$.

根据依赖关系的传递性可得:$\forall p\in P_{k-1}, f\dashrightarrow p$.

结合已知假设可得:$\forall i,k-1\leqslant i\leqslant N,\forall p\in P_i, f\dashrightarrow p$.

综上所述,可得:$\forall i,1\leqslant i\leqslant N,\forall p\in P_i, f\dashrightarrow p$. □

定理 1(可纳排序的充分条件). 给定规划问题 $Q=(F,A,I,G)$,令 $e,f\in G$ 分别是两个子目标原子.若 $f\nrightarrow e$,则 $e<_{AO}f$.

证明:根据约定,我们总是假设所讨论的规划问题满足可纳排序定义的第 1 个条件成立.

现在证明第 2 个条件也成立.即$\forall s\in S_{[e,\neg f]},\forall p\in OPT_{s\to[f]},\neg\exists a\in A(p), e\in del(a)$.

利用反证法证明.假设存在最优规划 $p\in OPT_{s\to[f]}$,令 $p=\langle a_1,\ldots,a_N\rangle$,满足:$\exists k,1\leqslant k\leqslant N, a_k\in A(p)\wedge e\in del(a_k)$.根据经典规划的约定,动作中的负效果原子必须也是该动作前提表的一个原子,即有:$\exists k,1\leqslant k\leqslant N, a_k\in A(p)\wedge e\in pre(a_k)$.于是,由引理 3 可知:$f\dashrightarrow e$.

而这与已知条件 $f\nrightarrow e$ 矛盾,因此假设不成立,原命题得证. □

定理 1 给出了可纳排序成立的一个充分条件.可以看出,只要两个子目标原子具有非依赖关系,它们就存在可纳排序关系,这为我们有效计算可纳排序提供了一个途径.下文将介绍原子依赖图技术,能够在多项式时间内近似计算可纳排序关系.

2.2.3 计算可纳排序关系和可纳子目标序列的算法

本节首先介绍原子依赖图,利用它能够在多项式时间内计算任意两个目标原子之间的依赖关系或非依赖关系.随后,我们将给出计算可纳子目标序列的算法.

(1) 原子依赖图

定义 5(原子依赖图). 已知规划问题 $Q=(F,A,I,G)$.关于 Q 的原子依赖图是一个有向图 $G_Q=(V,E)$,其中,顶点集合 $V=\{u|u\in F\}$,弧集合 $E=\{\langle u,v,a\rangle|u,v\in F$ 且 $u\to_a v, u$ 和 v 分别是弧的起点和终点,a 是弧的标签$\}$.

在不引起歧义的情况下,我们将弧$\langle u,v,a\rangle$简记为$\langle u,v\rangle$.我们称图中的顶点为原子顶点,称弧为直接依赖弧.如果在原子依赖图中存在一条从 u 到 v 路径$\langle u,\ldots,v\rangle$,则称从 u 到 v 是可达的;否则,称 u 到 v 不可达.

构造关于规划问题 Q 的原子依赖图是一个很简单的过程.构造过程仅需要对每个原子扫描一次,加入到顶

点集合中.然后对每个动作扫描一次,获取每个动作的增加表和前提表,然后将每对直接依赖关系$\langle u,v,a\rangle$($u\in add(a)$且 $v\in pre(a)$)加入到依赖图的弧集合中.假设规划问题 Q 中一共有$|F|$个原子,有$|A|$个动作,每个动作的增加表中最多有 N_{add} 个正效果(原子),最多有 N_{pre} 个前提(原子),那么构造过程的时间复杂度为

$$O(|F|+|A|\times N_{add}\times N_{pre}).$$

根据原子依赖图,我们可以很容易地得到原子顶点之间的依赖关系.

命题 3(直接依赖关系). 已知规划问题 Q 及其原子依赖图 $G_Q=(V,E)$,原子顶点 $u,v\in V$.u 通过 a 直接依赖于 v,当且仅当存在一条弧$\langle u,v,a\rangle\in E$.

命题 4(依赖关系). 已知规划问题 Q 及其原子依赖图 $G_Q=(V,E)$,原子顶点 $u,v\in V$.u 依赖于 v,当且仅当存在一条从 u 到 v 的可达路径.

命题 5(非依赖关系). 已知规划问题 Q 及其原子依赖图 $G_Q=(V,E)$,原子顶点 $u,v\in V$.u 非依赖于 v,当且仅当不存在任何一条从 u 到 v 的可达路径.

结合原子依赖图和依赖关系的定义可知,上述命题显然成立.

由上述命题易知,原子之间的依赖关系与原子依赖图中相应的原子顶点之间的可达关系一一对应.因此,计算任意两个原子之间的依赖关系或者非依赖关系,需要对原子依赖图进行全局遍历,以判断相应的原子顶点之间的可达性.如果具有可达路径,则说明存在依赖关系,否则是非依赖关系.如果采用深度优先或者广度优先搜索,由经典的图论知识不难得知,可达性计算具有关于图的顶点数和弧数的多项式时间复杂度.确切地说,复杂度为 $O(|V|^2)$(采用邻接矩阵存储方式)或者 $O(|V|+|E|)$(采用邻接表存储方式),V 和 E 分别是原子依赖图的顶点数和弧数.根据第 2.2.2 节关于可纳排序的充分条件,由于计算两个原子之间的非依赖关系具有多项式时间复杂度(顶点数和弧数),这说明能够在多项式时间内近似计算任意两个原子之间是否存在可纳排序关系.

(2) 计算可纳子目标序列

给定一个规划目标,我们首先需要依据目标原子之间是否具有可纳排序关系将它们划分到相应的子目标,然后对子目标进行排序,形成可纳子目标序列.由第 2.2.1 节表 2 关于两个原子之间依赖关系的组合情况易知,如果两个原子之间是相互独立关系或者单向依赖关系(不具有相互依赖关系),那么它们一定存在非依赖关系,也就会存在可纳排序关系(根据可纳排序的充分条件).因此,不具有相互依赖关系的原子应该被划分到不同的子目标中,而具有相互依赖关系的原子则被划分到同一个子目标中.这样就能得到尽可能多的可纳排序关系(子目标数量).然后,根据不同子目标之间的可纳排序关系进行排序,得到一个全序排列的具有可纳顺序关系的子目标序列.下面我们将详细介绍划分和排序子目标的算法.

在介绍可纳子目标序列算法之前,我们首先回顾相关的有向图和有向无环图的背景知识.如果一个有向图不包含平行弧(两个顶点之间同方向的弧)和自环,则称为简单图.对于一个简单有向图,如果图中任意一对顶点之间都是相互可达的,则称这个图是强连通的.具有强连通性质的最大子图称为强分图,或者称为强连通分支.

假设一个简单有向图 G,它有 N 个强分图,分别记为 $G_1,\ldots,G_N$,$1\leqslant N$.

性质 1. 如果在强分图 G_i 和 G_j 中($1\leqslant i,j\leqslant N$ 且 $i\neq j$)分别有两个顶点 s 和 t,并且从 s 到 t 可达,那么对任意顶点 $s'\in G_i$ 和顶点 $t'\in G_j$,s'到 t'可达,并且 t'到 s'不可达.

证明:由于 s 和 s'都在强分图 G_i 中,因此它们是相互可达的.又因为从 s 到 t 可达,即存在一条从 s 到 t 的路径,那么从 s'到 t 也是可达的.又因为 t 和 t'都在强分图 G_j 中,它们是相互可达的,因此从 s'到 t'仍然可达.

但是,由于 s 和 t 分别在不同的强分图中,因此从 t 到 s 不可达(否则,s 和 t 相互可达,就会在同一强分图中).同理,从 t'到 s'也不可达. □

如果强分图 G_i 和 G_j 分别有两个顶点 s 和 t,并且从 s 到 t 可达,那么它们满足性质 1,则称从 G_i 到 G_j 可达,而从 G_j 到 G_i 不可达.根据这一性质,可以构造一个新的有向图,称为强分顶点图.首先将每个强分图看成一个节点,称为强分顶点,分别记为 $V_1,\ldots,V_N$,$1\leqslant N$.于是,新的有向图可记为 $CG=(CV,CE)$,其中,$CV=\{V_i|V_i$ 是对应于 G_i 的强分顶点,$1\leqslant i\leqslant N\}$,$CE=\{\langle V_i,V_j\rangle|$从 G_i 到 G_j 可达$\}$.

性质 2. CG 是一个有向无环图.

证明:假设 CG 包含环路,即存在两个顶点 $V_i,V_j\in CV,i\neq j$,它们之间存在回路,也就是说,G_i 到 G_j 相互可达.但是,由于 G_i 和 G_j 是不同的强分图,由性质 1 可知,它们不可能相互可达.得出矛盾.说明假设不成立.得证. □

有向无环图的一个重要特点是可以进行拓扑排序.假设 $(V'_1,...,V'_N)$ 是对 CG 进行拓扑排序后的顶点序列.

性质 3. $\forall i,\forall j,1\leqslant i<j\leqslant N,\langle V'_j,V'_i\rangle\notin CE$.

证明:假设 $\langle V'_j,V'_i\rangle\in CE$.根据拓扑排序算法的特点,仅当某个顶点入度为 0 时,才会加入到拓扑序列中.由于 $i<j$,说明 V'_i 在 V_j'之前加入拓扑序列.也就是说,当 V'_i 加入拓扑序列时(入度为 0),V'_j 仍然在图中.显然,这与 $\langle V'_j,V'_i\rangle\in CE$ 矛盾.假设不成立.得证. □

图 2 给出了计算可纳子目标序列的算法伪代码.

Algorithm. *ComputeAdmissibleSubgoalSequence*(Q).
Input: A planning problem $Q=(F,A,I,G)$.
Output: Subgoals sequence $(SG_1,\ldots,SG_N)$ based on admissible orderings.
1. $t\leftarrow 1$;
2. while $G\neq\varnothing$ do
3. $G_t\leftarrow$randomly choose an element $e\in G$; //Initialize subgoal G_t with e
4. foreach $f\in G$ and $e\neq f$ do
5. if $(e\rightarrow f)\wedge(f\rightarrow e)$ then
6. $G_t\leftarrow G_t\cup\{f\}$; $G\leftarrow G\backslash\{f\}$;
7. $G\leftarrow G\backslash\{e\}$; $t\leftarrow t+1$;
8. $N\leftarrow t-1$;
9. $CG=(CV,CE)$ with $CV\leftarrow\{V_i|V_i$ is a vertex representing $G_i,1\leqslant i\leqslant N\}$ and $CE\leftarrow\varnothing$.
10. for $i\leftarrow 1$ to N do
11. for $j\leftarrow i+1$ to N do
12. randomly choose an element $e\in G_i$ and an element $f\in G_j$;
13. if $(e\rightarrow f)$ then $CE\leftarrow CE\cup\{\langle V_i,V_j\rangle\}$;
14. else if $(f\rightarrow e)$ then $CE\leftarrow CE\cup\{\langle V_j,V_i\rangle\}$;
15. $(V'_1,\ldots,V'_N)\leftarrow$Topological Sort for CG; //A topological sequence after sort
16. for $i\leftarrow 1$ to N do
17. $SG_i\leftarrow G_{id(i)}$; //Suppose V'_i corresponds to $G_{id(i)}$
18. return $(SG_1,\ldots,SG_N)$;

Fig.2 Algorithm for computing subgoals sequence based on admissible ordering relation

图 2 计算基于可纳排序关系的子目标序列的算法

给定一个规划领域 $Q=(F,A,I,G)$,算法 ComputeAdmissibleSubgoalSequence 分 3 个步骤来求解可纳子目标序列:(1) 按照相互依赖关系,将目标 G 划分为多个子目标(第 1 行~第 8 行);(2) 根据划分后的子目标及其相互(依赖)关系构造一个有向无环图 CG,然后利用有向无环图的性质对 CG 进行拓扑排序,从而得到一个全序排列(第 9 行~第 15 行);(3) 根据子目标与 CG 中的元素对应关系将子目标按照拓扑排序结果进行重新排列,得到可纳子目标序列(第 16 行~第 18 行).

在第 1 行,t 表示将要划分的子目标的索引,初始化为 1.当目标集合非空时(第 2 行),每次迭代(第 3 行~第 7 行)都将创建并维护一个新的子目标 G_t.算法首先任意选取一个目标集合中的原子 e 加入到子目标 G_t(第 3 行),此时,G_t 初始化为只包含原子 e.在第 4 行~第 6 行的循环中,算法将逐个判断目标集合中的其他原子是否与 e 具有相互依赖关系,如果是,则将该原子加入到子目标 G_t,并从目标集合中删除它(第 6 行).因此,G_t 中的所有原子都与 e 具有相互依赖关系.由于依赖关系的传递性,G_t 中的每对原子都具有相互依赖关系.并且,目标集合 G 中所有与 e 具有相互依赖关系的原子都会在中 G_t.当一次迭代结束时,将从目标集合 G 中删除 e,子目标索引 t 自加 1(第 7 行).当目标集合 G 都划分完毕后,将得到 $t-1$(记为 N)个子目标 $G_1,\ldots,G_N$(第 8 行).

第 9 行首先将初始化一个有向图 CG,按照下标的关系,每个顶点 $V_i(1\leqslant i\leqslant N)$都代表一个唯一的子目标 G_i(下标相同).随后,在第 10 行~第 14 行,CG 的弧集合 CE 将逐步更新.对于任意 2 个顶点 V_i 和 V_j,如果它们所对应的子目标 G_i 和 G_j 具有单向依赖关系(即 G_i 和 G_j 中的原子具有单向依赖关系),则相应的弧$\langle V_i,V_j\rangle$或者$\langle V_j,V_i\rangle$将加入到弧集合 CE.根据前面关于有向图的背景知识以及子目标的特点,CG 是一个有向无环图.在第 15 行,对 CG

进行拓扑排序,从而得到了一个拓扑有序序列 $(V'_1,...,V'_N)$.为方便讨论,我们假定 V'_i $(1\leqslant i\leqslant N)$与 $G_{id(i)}$一一对应.

最后,算法(第 16 行~第 18 行)将按照拓扑排序的结果,依次将对应的子目标赋值给可纳子目标序列(SG_1,…,SG_N)并返回.显然,可纳子目标序列中每个集合的每对原子之间都具有相互依赖关系.并且,在不同集合的原子之间一定存在非依赖关系,也就是可纳排序关系.

定理 2. 给定规划问题 $Q=(F,A,I,G)$.序列$(SG_1,\ldots,SG_N)$是图 2 所示算法返回的子目标序列,则有

$$\forall i,\forall j,1\leqslant i<j\leqslant N,\forall e\in SG_i,\forall f\in SG_j,e<_{AQ}f.$$

证明:根据算法第 9 行~第 14 行可知,CG 是一个有向无环图,根据拓扑排序的性质,有

$$\forall i,\forall j,1\leqslant i<j\leqslant N,\langle V'_j,V'_i\rangle\notin CE.$$

于是有如下推导过程:

$\forall i,\forall j,1\leqslant i<j\leqslant N,\langle V'_j,V'_i\rangle\notin CE$

$\Leftrightarrow\forall i,\forall j,1\leqslant i<j\leqslant N,\forall e\in G_{id(i)},\forall f\in G_{id(j)},f\nrightarrow e$ V'_i $(1\leqslant i\leqslant N)$与 $G_{id(i)}$一一对应

$\Leftrightarrow\forall i,\forall j,1\leqslant i<j\leqslant N,\forall e\in SG_i,\forall f\in SG_j,f\nrightarrow e$ 由算法第 16 行、第 17 行,$SG_i(1\leqslant i\leqslant N)$与 $G_{id(i)}$对应

$\Rightarrow\forall i,\forall j,1\leqslant i<j\leqslant N,\forall e\in SG_i,\forall f\in SG_j,e<_{AQ}f$ 根据可纳排序关系成立的充分条件

得证. □

从算法中也可以看出,由于拓扑排序输出的排序序列可能不是唯一的,因此可纳子目标序列也可能不是唯一的.

2.3 在规划中使用可纳子目标排序

本节将讨论如何在规划中使用可纳子目标排序.我们采用在 IPP[7]和 FF[2]规划器中使用的 GADP(goal agenda-driven planning)算法.

GADP 是一种迭代增量式的规划算法,如图 3 所示.给定一个可纳子目标序列 ASS:$(SG_1,\ldots,SG_N)$,算法(第 1 行~第 3 行)首先生成一个长度为 N 的目标议程(goal agenda entries):$(G_1,\ldots,G_N)$,并且 $G_1=SG_1,G_t=G_{t-1}\cup SG_t$, $2\leqslant t\leqslant N,G_N=G$.在第 4 行~第 8 行,算法采用基本规划器迭代求解每个子问题 Q_t(由当前初始状态 I_t 和子目标状态 G_t 组成,$1\leqslant t\leqslant N$).在每次规划迭代中,仅实现一个子目标 G_t,得到部分规划解 π_t,然后再把子目标实现后的状态作为新的初始状态 $I_{t+1}=exec(I_t,\pi_t)$,进入下一个子问题 Q_{t+1} 的规划过程,直到最后一个子目标 G_N 得以实现,整个规划过程结束.算法返回规划解$\langle\pi_1,\ldots,\pi_N\rangle$.

Algorithm. *GADP*(Q,*ASS*).
Input: A planning problem Q: (F,A,I,G); an admissible subgoal sequence *ASS*: $(SG_1,\ldots,SG_N)$.
Output: A solution plan of Q.
1. $G_1\leftarrow SG_1$;
2. for $t\leftarrow 2$ to N do
3. $\quad G_t\leftarrow G_{t-1}\cup SG_t$;
4. $I_1\leftarrow I$;
5. for $t\leftarrow 1$ to N do
6. $\quad Q_t\leftarrow(F,A,I_t,G_t)$;
7. $\quad \pi_t\leftarrow$solution plan returned by calling basic planner to solve Q_t;
8. $\quad I_{t+1}\leftarrow exec(I_t,\pi_t)$;
9. return $\langle\pi_1,\ldots,\pi_N\rangle$;

Fig.3 Goal agenda-driven planning algorithm with admissible subgoal ordering

图 3 在 GADP 算法中使用可纳子目标排序

GADP 算法对每个子目标的规划可以采用任意规划器(称为基本规划器).文献[7]已经证明,如果基本规划器能够保持正确性(soundness),由于 $G_N=G$,GADP 算法就是正确的.而且,如果规划问题可解并且不存在死锁(deadlock)状态,那么采用任何保持完备性的基本规划器,GADP 算法都能求得一个规划解,即它是完备的.当然,它并不能保证对任意规划问题都保持完备性.如果在求解某个子问题的迭代过程中基本规划器返回失败,那么我们将放弃这种增量式的规划算法,而采用完备性的规划算法求解原始规划问题(不再划分和排序子目标).

3 实验和分析

本文所述的可纳子目标排序思想及其相关方法已经在我们的规划系统 ASOP 实现.当前版本为 ASOPv1.0. ASOP 的核心由一个可纳子目标排序引擎(预处理模块)、一个增量式规划框架和一个基本规划器组成,如图 4 所示.可纳子目标排序引擎是基于第 2.2 节所介绍的算法,增量式规划框架是在第 2.3 节所介绍的 GADP 算法,基本规划器是用于求解每个规划子问题(由一个初始状态和一个子目标集合构成).理论上,任何能够处理 PDDL 语言的有效的规划器都可以作为 ASOP 的基本规划器.

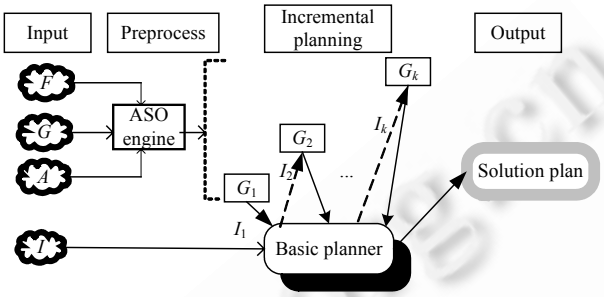


Fig.4 System architecture of ASOP planner

图 4 ASOP 规划器的系统架构

在本文中,我们将采用修改后的 FF(version 2.3)作为 ASOPv1.0 的基本规划器,.FF[2]是一个优秀的高性能开源规划器,在国际智能规划大赛 IPC-2 和 IPC-3 的 STRIPS 组中取得了优异的成绩,被研究人员广泛引用.FF 是一种前向启发式状态空间搜索的规划器,采用放宽规划图技术(relaxed planning graph)计算搜索状态的启发式估值,采用增强型爬山法(enforced hill climbing)作为快速前向启发式搜索算法,并定义有利动作(helpful actions)作为有效的剪枝策略.由于 EHC 是一种不完备的搜索算法,当 EHC 搜索失败时,FF 将转而采用完备的贪婪最佳优先算法对原始规划问题进行求解(不再划分和排序子目标),并禁止有利动作剪枝.由于 FFv2.3 本身包含了计算 RSO 子目标排序(reasonable subgoal ordering,以下称为 RSO 排序,见第 4 节的介绍)的功能模块,在将它集成到 ASOPv1.0 作为基本规划器时,我们去掉了该排序模块;而当单独使用 FFv2.3 进行实验时,则保持该排序模块不变.这样既能独立地测试可纳排序对规划效率的影响,也能公平地对比 ASOP 与 FFv2.3 的规划性能差异,也就能够公平地对比可纳排序与 RSO 排序对规划效率的影响.另外,如果 EHC 搜索失败,此时子目标排序功能将不再起作用,而采用完备性的规划算法求解原始规划问题.我们将会在实验数据中对失败情形(如果存在的话)进行标识.

我们采用了国际规划大赛 IPC-3,IPC-4 和 IPC-5 的 4 个标准测试领域的 STRIPS 语言版本作为测试数据.这 4 个测试领域分别是:IPC-3 的 ZenoTravel 领域、IPC-4 的 Satellite 领域、IPC-5 的 TPP 领域和 Rovers 领域.这些领域的测试问题数目分别为 20 个、36 个、30 个、40 个.我们并不列出全部问题的实验数据,仅给出每个领域较复杂的 20 个测试问题的实验数据.

测试平台为 Fedora Release-8(Linux kernel 2.6.23),编译器为 g++(GCC) 4.1.2,内存为 1GB,CPU 为 Intel Core Duo T2450 2.0GHz.对每个领域问题,设定规划超时限制为国际规划大赛规定的 30 分钟(1 800 秒).

首先给出在这些规划问题中存在的可纳子目标排序数目.其次,从以下 3 个方面全面对比 ASOP 与 FF 的性能:规划求解过程中所扩展的的节点数(即搜索状态的数目)、规划时间、规划解的长度.

表 2 给出了 4 个领域中较复杂的 20 个测试问题上所计算得到的可纳子目标排序数目.可以看出,在这些问题中存在着大量的可纳排序关系.因此,ASOP 能够充分利用可纳排序关系划分和排序目标集合原子,将大型复杂的规划问题划分为多个更简单的子问题.需要注意的是,我们并没有在表 2 中列出 FF 规划系统所求解到的 RSO 排序的数目,这是因为,除了 TPP 领域的部分问题中存在两个 RSO 排序关系外,其他领域的全部问题都不

存在 RSO 排序关系(即 RSO 的数目为 1).因此,我们将注意力放在本文所讨论的可纳排序关系上.

Table 2 Numbers of admissible subgoal orderings (#ASO) extracted by our approach

表 2 本文方法所提取的可纳子目标排序数目(#ASO)

ZenoTravel		Satellite		TPP		Rovers	
Problem	#ASO	Problem	#ASO	Problem	#ASO	Problem	#ASO
1	8	17	20	11	6	21	14
2	8	18	13	12	7	22	16
3	10	19	28	13	8	23	19
4	12	20	41	14	9	24	21
5	14	21	37	15	10	25	10
6	17	22	45	16	11	26	14
7	23	23	51	17	12	27	20
8	21	24	70	18	13	28	19
9	27	25	64	19	14	29	16
10	25	26	68	20	15	30	25
11	8	27	65	21	11	31	24
12	8	28	94	22	12	32	35
13	10	29	93	23	13	33	44
14	12	30	114	24	14	34	32
15	14	31	133	25	15	35	63
16	17	32	182	26	16	36	41
17	23	33	231	27	17	37	58
18	21	34	141	28	18	38	55
19	27	35	187	29	19	39	63
20	25	36	178	30	20	40	69

图 5 分别给出了 4 个领域在规划搜索中所扩展的节点数和规划求解时间的实验结果对比.

在 ZenoTravel 领域中,由于问题 1~问题 10 规模比较小,因此无论是所扩展的节点数还是规划求解时间,ASOP 都与 FF 接近.而在问题 11~问题 20 中,随着规模的增加,规划问题也越来越复杂,ASOP 的规划求解过程(搜索过程)比 FF 明显要快.并且在问题 17~问题 20 中,ASOP 的规划求解时间和扩展的节点数都比 FF 少 1 个数量级,也就是说,规划效率比 FF 要高 1 个数量级.

在 Satellite 领域,ASOP 所扩展的节点数和规划求解时间都比 FF 要少.并且在大规模的测试问题上,如问题 20、问题 23~问题 36 等,ASOP 的规划求解效率比 FF 高 1 个数量级甚至高 2 个数量级.

在 TPP 领域,ASOP 所扩展的节点数和规划求解时间都比 FF 要少.在问题 19~问题 28 上(除问题 22 外),ASOP 的规划求解时间和扩展的节点数都比 FF 少两个数量级.而在问题 29、问题 30,FF 在 30 分钟内依然无法返回规划解,而 ASOP 仅在 15 秒钟内就求解成功.

在 Rovers 领域同样能够看出,ASOP 所扩展的节点数和规划时间都比 FF 要少.尤其在大规模的问题 32~问题 40 上,ASOP 的规划求解时间和扩展的节点数都比 FF 少 2 个数量级.

总结、分析以上的结果对比不难发现,由于在这些领域问题中存在大量的可纳(子目标)排序关系,ASOP 通过利用可纳排序关系对规划目标集合进行划分和排序,将复杂的规划问题分解为多个更简单的子问题,从而有效地降低了问题的规模和复杂程度;而依次求解各个子问题相对于一次性求解原始规划问题要简单得多,因此与 FF 规划器相比,ASOP 规划器能够极大地提高规划求解效率.

表 3 给出了 4 个领域的规划解长度的对比.可以看出,在 ZenoTravel 领域、TPP 领域和 Rovers 领域的大部分测试问题中,ASOP 的规划解长度比 FF 要长.而在 Satellite 领域的大部分问题中,ASOP 的规划解长度则比 FF 要短.这说明 ASOP 在优化规划解的质量方面仍然有待提高.造成这种状况的一个主要原因在于,我们所提取的可纳排序关系并非总是符合最优的子目标实现顺序,这将是我们未来需要重点改进的工作之一.

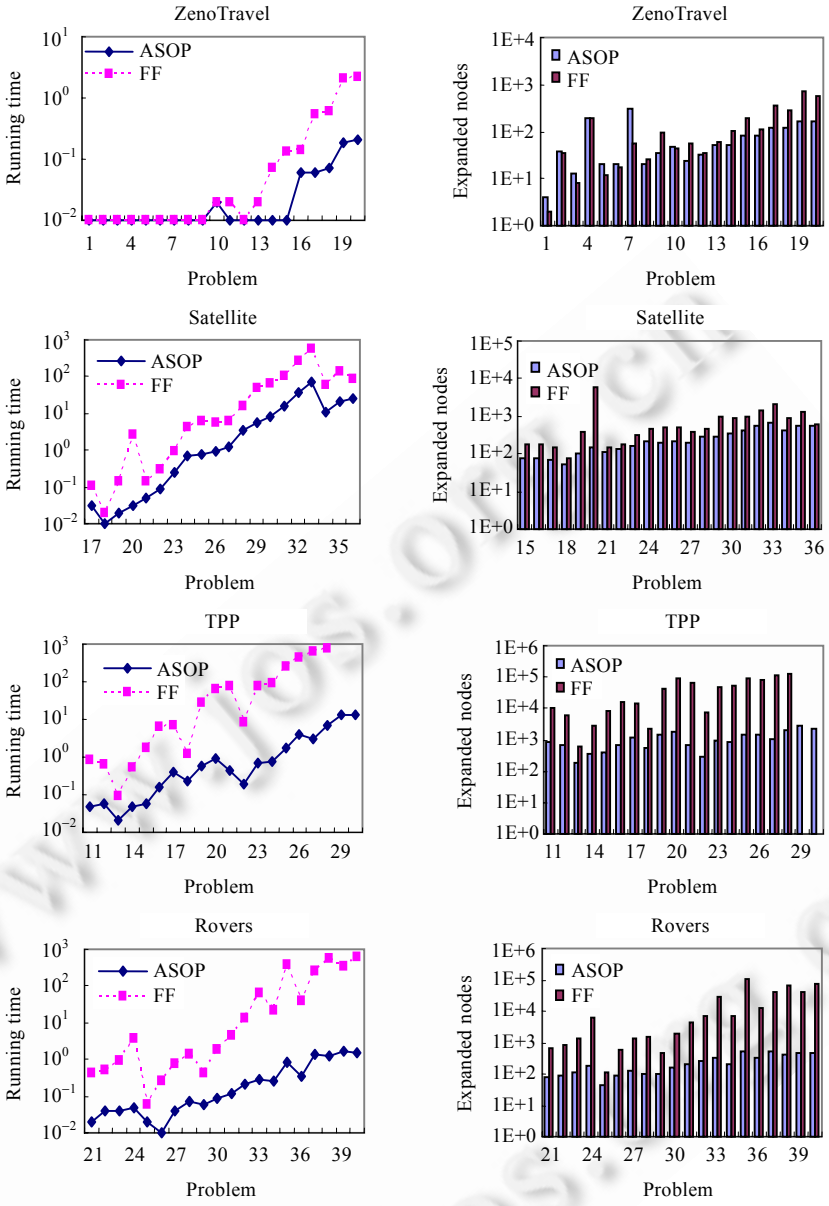


Fig.5 Performance comparison in terms of running time and expanded nodes between ASOP and FFv2.3

图 5 ASOP 和 FFv2.3 的性能对比(运行时间与扩展节点数)

4 相关工作

有关规划问题的划分和排序研究已有较长的历史,相关工作包括文献[4,7−16].本节讨论与子目标排序密切相关的一些研究成果和最新进展.

Koehler 和 Hoffmann 等人[7]提出了 Forced Subgoal Ordering(简称 FSO 排序)和 Reasonable Subgoal Ordering(简称 RSO 排序);随后,Hoffmann,Porteous 和 Sebastia 等人[8,9]又进一步提出了 Weakly Reasonable Ordering(简称 WRO 排序).关于两个子目标 e 和 f 的 FSO 排序 $e<_{FSO}f$ 是指,如果先实现了 f 且 e 未实现,那么 e

永远无法实现,所以需要强制性地先实现 e 再实现 f.关于两个子目标 e 和 f 的 RSO 排序 $e<_{RSO}f$ 是指,如果先实现了 f 且 e 未实现,那么任何能够实现 e 的规划一定会删除 f.因此,合理的顺序应该是先实现 e 再实现 f.关于两个子目标 e 和 f 的 WRO 排序 $e<_{WRO}f$ 与 RSO 排序的定义类似,但它仅限于在实现 f 但未实现 e 的局部最优规划生成的可达状态下,任何能够实现 e 的规划一定会删除 f.实际上,这 3 种顺序关系都是一类强约束的顺序关系,它们要求子目标必须按照给定的顺序实现(即先实现 e 再实现 f);否则,将无法求解规划(例如 FSO 排序),或者 f 实现之后,由于临时性地阻碍了 e 的实现,f 又会被再次删除.这样的顺序关系在大部分规划领域问题都极少存在.而我们的可纳排序则弱化了子目标之间的约束,它给出了一种建议性的合理而有效的排序机制,但它并不强制任意规划过程都必须按照这种子目标顺序实现.由实验可知,可纳排序关系在很多规划问题中存在,因此它能够有效地求解很多大规模的规划问题.

Table 3 Comparison of solution plan length between ASOP and FFv2.3

表 3 ASOP 和 FFv2.3 的规划解长度的对比

ZenoTravel			Satellite			TPP			Rovers		
Problem	ASOP	FF	Problem	ASOP	FF	Problem	ASOP	FF	Problem	ASOP	FF
1	1	1	17	49	48	11	125	109	21	60	63
2	6	6	18	35	35	12	121	101	22	68	70
3	7	6	19	67	73	13	73	61	23	80	81
4	11	11	20	99	107	14	112	101	24	129	127
5	14	11	21	77	77	15	129	113	25	32	32
6	15	12	22	91	92	16	172	148	26	65	61
7	16	16	23	107	107	17	153	139	27	98	98
8	12	12	24	139	141	18	132	122	28	77	74
9	27	24	25	127	133	19	203	179	29	71	56
10	35	24	26	141	144	20	246	226	30	124	116
11	15	14	27	135	138	21	228	221	31	137	130
12	25	23	28	193	200	22	167	145	32	183	159
13	38	28	29	195	195	23	223	199	33	246	240
14	37	34	30	228	235	24	217	195	34	155	145
15	61	41	31	272	274	25	261	235	35	391	380
16	60	45	32	369	371	26	262	236	36	247	220
17	93	69	33	469	473	27	283	252	37	378	347
18	90	62	34	290	291	28	304	274	38	294	273
19	118	83	35	383	382	29	397	–	39	373	340
20	124	90	36	359	365	30	382	–	40	352	313

Chen 等人[4]提出了一种子目标划分和求解的策略,将时态规划问题中的互斥约束根据所对应的子目标进行划分,并独立求解每个子目标,最后再解决全局的违背约束,得到最终的规划解.该思想应用在时态规划器 SGPLAN4 以及 SGPLAN5,并且求解每个子目标的基本规划器是 Metric-FF.在 Chen 的工作中,子目标的划分非常简单,每个目标集合中的原子就是一个独立的子目标,并且没有排序过程.这与我们的工作不尽相同.

Hsu 等人[10]提出了一种改进的放宽式子目标排序算法(relaxed-plan ordering algorithm),并分析了子目标划分的粒度控制问题.这种放宽式的排序算法利用了放宽规划容易求解的特点,在构建放宽规划之后提取出所谓的放宽排序关系(relaxed-plan ordering relation).在 IPC-4 的很多领域问题中能够得到比 RSO 排序更多的子目标顺序关系,然而放宽排序关系缺乏形式化的定义,没有从理论上说明该排序的合理性.并且,由于放宽式规划忽略动作负效果,从中提取出的排序关系可能存在很多错误的顺序关系,需要对这些错误排序进行动态的调整.而我们的可纳排序关系具有严格的形式化定义及其理论上的合理性.

另一种与子目标排序关系非常类似的是对非子目标进行的排序,称为 landmark ordering,即里程碑排序.landmark 的概念最早由 Hoffmann 等人[8,9]提出,它是指在任何一个规划解中,一定会在某个中间状态出现的非子目标原子.文献[8,9]中给出了求解 landmark 的算法,并且能够改善采用前向启发式搜索或者图规划方法的规划器的性能.Richter 和 Helmert 等人[11]则提出了使用 landmark 生成伪启发式并和搜索框架中的其他启发式结合的方法,从而提高启发式搜索规划器的搜索成功率以及规划解的质量.将可纳排序关系推广到对非子目标进行排序,这也将是我们未来的研究工作之一.

吴向军等人[12]通过对 STRIPS 规划领域中的动作效果关系的研究,提出了谓词之间的的排斥性阻碍关系,

并利用排斥性阻碍关系对规划子目标进行排序.但是,该文献并没有给出关于该排序的形式化定义和性质,仅是对特定领域(例如 Block 领域)的一种常识性的推理判断.因此,它实际上是一种领域相关的排序方式.这与我们具有明确严格形式化定义并且能够应用在任何领域(即领域无关)的排序方式是不同的.另外,该文献仅仅初步论述了在排斥性阻碍关系存在的情况下如何获取子目标之间的顺序关系,对其他情况下如何对子目标进行排序并没有讨论,并且也没有分析和比较子目标排序对规划效率和规划质量的影响.

采用子目标排序的规划器还包括 DTG-Plan[17],LAMA[18]等.文献[19]则研究了一种利用状态不变式提取目标之间顺序关系的方法,比 Koehler 等人[7]的算法具有一定的改进.

5 结束语

求解大规模复杂的规划问题仍然是当前规划研究的难点和热点.本文关注于采用子目标排序的方法来降低问题规模和复杂程度,我们提出了一种称为可纳子目标排序的子目标排序关系,给出了计算可纳排序的有效方法.利用可纳排序关系对规划目标进行划分和排序,使得复杂的规划问题能够被分解成更简单的子问题,有助于提高规划求解的效率.在国际规划大赛的标准测试问题上的实验结果表明,我们的方法能够有效地提取出可纳排序关系,并极大地改善规划性能.在大规模复杂的规划问题中,ASOP 的规划效率能够比国际规划大赛上的顶级高性能规划系统 FFv2.3 提高 1~2 个数量级,这显示了我们的理论和方法具有强大的生命力.

今后,我们将研究如何在可纳排序关系的框架下改善子目标排序结果,以期优化规划解的质量.另外,改进当前的增量式规划算法,以利用可纳排序的优点进行搜索空间的约减,从而提高规划性能,也是我们未来的工作之一.同时,我们也将扩展可纳排序的定义范畴,使其能够支持表达能力更为丰富的规划语言,如包含条件效果的 ADL 语言等.

References:

[1] Bylander T. The computational complexity of propositional STRIPS planning. Artificial Intelligence, 1994,69(1-2):165−204. [doi: 10.1016/0004-3702(94)90077-9]

[2] Hoffmann J, Nebel B. The FF planning system: Fast plan generation through heuristic search. Journal of Artificial Intelligence Research, 2001,14:253−302. [doi: 10.1613/jair.855]

[3] Gerevini A, Saetti A, Serina I. Planning through stochastic local search and temporal action graphs in LPG. Journal of Artificial Intelligence Research, 2003,20:239−290.

[4] Chen Y, Wah BW, Hsu CW. Temporal planning using subgoal partitioning and resolution in SGPlan. Journal of Artificial Intelligence Research, 2006,26:323−369.

[5] Helmert M. The fast downward planning system. Journal of Artificial Intelligence Research, 2006,26:191−246. [doi: 10.1007/s10462-007-9049-y]

[6] Chen Y, Huang R, Xing Z, Zhang W. Long-Distance mutual exclusion for planning. Artificial Intelligence, 2009,173(2):365−391. [doi: 10.1016/j.artint.2008.11.004]

[7] Koehler J, Hoffmann J. On reasonable and forced goal orderings and their use in an agenda-driven planning algorithm. Journal of Artificial Intelligence Research, 2000,12:338−386.

[8] Hoffmann J, Porteous J, Sebastia L. Ordered landmarks in planning. Journal of Artificial Intelligence Research, 2004,22:215−278.

[9] Porteous J, Sebastia L, Hoffmann J. On the extraction, ordering, and usage of landmarks in planning. In: Cesta A, Borrajo D, eds. Proc. of the Europearn Conf. on Planning 2001 (ECP 2001). Toledo: Springer-Verlag, 2001. 37−48.

[10] Hsu CW, Wah BW, Chen Y. Subgoal ordering and granularity control for incremental planning. In: Proc. of the IEEE Int'l Conf. on Tools with Artificial Intelligence 2005 (ICTAI 2005). Hong Kong: IEEE Computer Society, 2005. 507−514. [doi: 10.1109/ICTAI.2005.118]

[11] Richter S, Helmert M, Westphal M. Landmarks revisited. In: Fox D, Gomes CP, eds. Proc. of the AAAI 2008. Chicago: AAAI Press, 2008. 975−982.

[12] Wu XJ, Jiang YF, Ling YB. Research on relations of effect of action for STRIPS domain. Journal of Software, 2007,18(6): 1328−1349 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/1328.htm [doi: 10.1360/jos181328]

[13] Korf RE. Macro-Operators: A weak method for learning. Artificial Intelligence, 1985,26(1):35−77. [doi: 10.1016/0004-3702(85)90012-8]

[14] Knoblock CA. Automatically generating abstractions for planning. Artificial Intelligence, 1994,68(2):243−302. [doi: 10.1016/0004-3702(94)90069-8]

[15] Koehler J. Solving complex planning tasks through extraction of subproblems. In: Simmons RG, Veloso MM, Smith S, eds. Proc. of the 4th Int'l Conf. on Artificial Intelligence Planning Systems (AIPS'98). Pittsburgh: AAAI Press, 1998. 62−69.

[16] Irani K, Cheng J. Subgoal ordering and goal augmentation for heuristic problem solving. In: McDermott JP, ed. Proc. of the Int'l Joint Conf. on Artificial Intelligence 1987 (IJCAI'87). Milan: Morgan Kaufmann Publishers, 1987. 1018−1024.

[17] Chen Y, Huang R, Zhang W. Fast planning by search in domain transition graph. In: Fox D, Gomes CP, eds. Proc. of the AAAI 2008. Chicago: AAAI Press, 2008. 886−891.

[18] Richter S, Westphal M. The LAMA planner: Guiding cost-based anytime planning with landmarks. Journal of Artificial Intelligence Research, 2010,39:127−177.

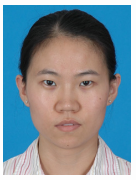
[19] Li Y, Jin Z. Goal ordering extraction and abstract method. Journal of Software, 2006,17(3):349−355 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/349.htm [doi: 10.1360/jos170349]

附中文参考文献:

[12] 吴向军,姜云飞,凌应标.STRIPS 规划领域中动作效果关系的研究.软件学报,2007,18(6):1328−1349. http://www.jos.org.cn/1000-9825/18/1328.htm [doi: 10.1360/jos181328]

[19] 李颖,金芝.目标间顺序关系的提取及其抽象方法.软件学报,2006,17(3):349−355. http://www.jos.org.cn/1000-9825/17/349.htm [doi: 10.1360/jos170349]

梁瑞仕(1982－),男,湖南耒阳人,博士,讲师,主要研究领域为智能规划,启发式搜索.



边芮(1982－),女,博士,讲师,主要研究领域为智能规划.

姜云飞(1945－),男,教授,博士生导师,主要研究领域为自动推理,智能规划,基于模型的诊断.

吴向军(1965－),男,博士,副教授,主要研究领域为人工智能,算法设计,网络应用.