

路标计数启发式引导的分解规划方法*

魏 唯^{1,2,3}, 欧阳彤^{1,2}, 吕 帅^{1,2}

¹(吉林大学 计算机科学与技术学院, 吉林 长春 130012)

²(符号计算与知识工程教育部重点实验室(吉林大学), 吉林 长春 130012)

³(吉林大学 公共计算机教学与研究中心, 吉林 长春 130012)

通讯作者: 欧阳彤, E-mail: ouyd@jlu.edu.cn

摘 要: 路标信息能够准确描述智能规划问题解空间的基本形态, 提出由路标信息引导的分解规划方法, 求解过程由路标计数启发式引导增强爬山算法向目标方向进行, 根据路标的完成情况分段求出规划解. 从全局范围上看, 爬山过程逐渐实现更多的路标, 路标计数启发式估值的降低引发规划任务的分解, 当搜索过程遇到估值更低的状态时, 提取一段爬山路径. 如此反复执行“搜索-提取”过程, 直至路标计数启发式的估值降低为 0, 各段爬山路径构成最终的规划解. 采用最新国际通用的标准测试问题进行实验测试, 结果表明: 由路标计数启发式引导的分解规划方法能够更好地发挥路标信息的优势, 实现了搜索范围的压缩, 可更快地生成规划解.

关键词: 路标计数启发式; 增强爬山; 分解规划方法; 爬山路径

中图法分类号: TP181 **文献标识码:** A

中文引用格式: 魏唯, 欧阳彤, 吕帅. 路标计数启发式引导的分解规划方法. 软件学报, 2013, 24(10): 2327-2339. <http://www.jos.org.cn/1000-9825/4383.htm>

英文引用格式: Wei W, Ouyang DT, Lü S. Decomposed planning guided by landmark counting heuristic. Ruan Jian Xue Bao/ Journal of Software, 2013, 24(10): 2327-2339 (in Chinese). <http://www.jos.org.cn/1000-9825/4383.htm>

Decomposed Planning Guided by Landmark Counting Heuristic

WEI Wei^{1,2,3}, OUYANG Dan-Tong^{1,2}, LÜ Shuai^{1,2}

¹(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

²(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (Jilin University), Changchun 130012, China)

³(Center for Computer Fundamental Education, Jilin University, Changchun 130012, China)

Corresponding author: OUYANG Dan-Tong, E-mail: ouyd@jlu.edu.cn

Abstract: Landmarks can capture the features of the solution space of planning tasks precisely. In this paper, a decomposed planning method guided by landmark information is proposed. The method executes an enforced hill-climbing procedure guided by the landmark-counting heuristic towards the goal, searching for a plan along with completions of the landmarks. Globally the hill-climbing procedure achieves the landmarks one after another. A decrease in the landmark counting heuristic estimation causes task decomposition and whenever the search encounters a state with a lower estimation, a hill-climbing fragment is extracted. Such “search-extract” procedure is repeated until the estimation of the landmark counting heuristic decreases to zero eventually, and then all the extracted fragments are connected into the final plan. Experiment results show that the decomposed planning method guided by landmark-counting heuristic makes use of the landmark information in a more flexible way, usually cutting down the search space dramatically and find the plan much faster.

Key words: landmark-counting heuristic; enforced hill-climbing; decomposed planning; hill-climbing fragment

* 基金项目: 国家自然科学基金(61272208, 61133011, 60973089, 61003101, 61170092, 61300049); 吉林省科技发展计划(20101501, 20100185, 201101039); 国家教育部博士点专项基金(20100061110031, 20120061120059); 博士后科学基金面上资助项目(2011M 500612); 浙江省自然科学基金(Y1100191); 浙江师范大学计算机软件与理论省级重中之重学科开放基金(ZSDZZZXK12)

收稿时间: 2012-07-04; 修改时间: 2012-10-29; 定稿时间: 2013-02-04

智能规划是人工智能研究的一个重要领域,智能规划技术已被广泛应用于社会生活的许多方面,取得的成果是显著的^[1,2].在过去十多年中,智能规划的研究取得了巨大的突破,无论在求解规模上还是求解效率上都有数量级的提高^[3-6].基于启发式搜索的规划方法是当前规划研究的热点,在国际智能规划竞赛中,基于启发式搜索的规划系统多次取得优异的成绩,各种启发式函数相继涌现,其中比较有代表性的包括:和/最大启发式^[7]、基于放松规划解的启发式(也常被称为 FF 启发式)^[8]、因果图启发式^[9]以及加强上下文的和启发式^[10]等.

近年来,路标被广泛应用于智能规划领域.路标(landmark)的概念最初由文献[11]提出,泛指在每条解路径上都必须成立的命题.后来,研究者相继提出了析取命题路标、合取命题路标、动作路标等多种路标形式.路标及路标之间的排序关系用一个有向图表示,称为路标图,对路标图进行分析,能够为规划求解过程提供帮助信息.文献[12]根据路标排序关系将原始规划任务进行分解,各个路标指定为子任务需要到达的目标,这种方法在求解过程中提示规划算法接下来将要搜索的阶段性目标,帮助规划算法做出下一步选择,在一些域上,该方法能够加快求解过程.然而,强制指定搜索的中间目标通常得到质量较差的规划解,甚至导致搜索失败.路标信息的另一个典型应用是启发式函数的设计,近年来,基于路标的启发式包括路标计数启发式^[13]、代价最优启发式^[14]、路标切割启发式^[15]等.路标计数启发式(landmark-counting heuristic)是最直接的设计方法,采用从当前状态到达目标还需要实现的路标个数 $l=n-m+k$ 作为状态估值,其中, n 是给定问题提取出的路标总数, m 为当前已经到达的路标个数, k 为已到达但还需要再次实现的路标个数.以此设计出的规划系统 LAMA 于 2008 年和 2011 年连续两次获得了“国际规划比赛连续时间满意规划组”的冠军,路标启发式的成功使它成为了新的研究热点^[16].

在求解不同类型的规划任务时,各种启发式分别拥有各自的优势,Fast Downward 规划系统首次提出搜索过程中交替使用两个启发函数,选用的是因果图启发式和著名的 FF 启发式.LAMA 规划系统建立在 Fast Downward 规划系统^[9]的核心框架之上,继承了多启发式搜索、延迟启发式估值、优先动作等搜索策略,使用路标计数启发式和 FF 启发式,搜索过程在两个启发式之间切换,同时发挥两个启发函数的指导能力.目前,越来越多的规划系统采用多个启发式引导搜索过程.

从 2008 年国际规划比赛开始,规划测试问题中增加了处理动作执行代价的需求,对规划系统性能的评判标准主要侧重于规划解的质量,即规划解中各个动作的执行代价之和,求解时间限定在 30 分钟之内.对于这类规划任务,如果要获得高质量的规划解,启发函数的设计需要将目标的实现代价考虑在内,指导规划算法寻找代价最优的规划解.然而,搜索过程一味地追求代价的最低很可能减慢求解速度,在 30 分钟内无法给出任何规划解.不同于“最优规划组”的规定,“可满足规划组”并不严格要求规划解的最优性,而更加注重规划算法的求解能力,希望对于大多数问题都能够在尽量短的时间内给出一个可用的规划解,因此可满足规划算法通常在规划解的质量与时间效率之间加以折中.为了充分利用给定的求解时间,LAMA 规划系统采用 Anytime 搜索机制,实现了两种搜索算法:贪心最好优先(greedy best-first)算法和加权 A*(weighted A*)算法,规划求解过程首先执行一次贪心最好优先搜索,目的是尽快完成求解任务,成功找到一个规划解,此后,在剩余时间内,基于 Anytime 的思想反复执行一系列加权 A*搜索,每一次迭代过程启发式估值的权重逐渐降低,目的是逐步提高规划解的质量.

可见,在 Anytime 机制下,规划算法能够在任意时刻给出当前已求得的最好规划解,因此,初始解的生成效率尤为重要.根据爬山算法的高效性,本文提出路标计数启发式引导下的分段求解思想,Anytime 的首次搜索过程采用的增强爬山搜索框架,以路标计数启发式作为评价函数,通过不断减少路标计数启发式的估值来到达目标状态.整个求解过程中,增强爬山算法与贪心最好优先算法嵌套使用,路标计数启发式与 FF 启发式相互配合.研究表明:本文提出的规划算法更能够发挥路标信息的指导能力,在搜索方向上更具指向性,生成初始规划解耗费的时间更少.

1 相关定义

1.1 经典规划问题

定义 1^[16]. 一个经典规划问题可以形式化地表示为一个五元组 $T=(F,A,I,G,C)$,其中,

- F 是有限逻辑命题的集合;

- A 是域中所有可用动作的集合,动作 $a \in A$ 是一个三元组 $\langle pre(a), add(a), del(a) \rangle$,其中 $pre(a), add(a)$ 和 $del(a)$ 均为原子命题的集合,分别表示动作 a 的前提条件、添加效果和删除效果;
- $I \subseteq F$ 为初始状态;
- $G \subseteq F$ 为目标条件;
- $C: A \rightarrow N_0^+$ 是动作集合 A 到非负整数集的映射,为每个动作指定执行代价.

一个状态 $s \subseteq F$ 是一组逻辑命题的集合,表示在这个状态下所有取值为真的命题.对于一个动作 a 和一个状态 s ,若 $pre(a) \subseteq s$,则称动作 a 可应用于状态 s .在状态 s 上应用一个动作 a 后,实施集合运算得到后继状态:

$$Result(s,a) = s - del(a) \cup add(a).$$

类似地,在状态 s 上执行动作序列 $\pi = \langle a_1, a_2, \dots, a_n \rangle$ 后得到的结果为 $Result(s,\pi) = Result(Result(s,a_1), \langle a_2, \dots, a_n \rangle)$.

定义 2^[6]. 对于给定的规划问题 T ,若存在一个动作序列 $\pi = \langle a_1, a_2, \dots, a_n \rangle$,满足 $G \subseteq Result(I,\pi)$,则称 π 是问题 T 的一个规划解,执行代价为 π 中各个动作执行代价之和 $\sum_{i=1}^n C(a_i)$.

1.2 路标计数启发式与FF启发式

路标及其排序关系的提取过程在搜索开始之前的预处理阶段完成,构造路标图.通常意义上,路标可以是任意形式的公式,但考虑到实际提取路标的计算量,目前,LAMA 规划系统使用的路标计数启发式仅考虑易于处理的命题路标与析取命题路标.路标图 $LG = \langle L, O \rangle$ 大致给出了规划解到达各个路标的先后次序以及哪些路标需要被多次实现,其中, L 为问题预处理阶段所提取出的路标集合, O 是所有形式为 $\varphi \rightarrow_x \psi$ 的路标排序关系集合.

路标 φ 在状态 s 上被实现的条件是 φ 在 s 上为真并且路标图中所有排序在 φ 之前的路标都已在路径 π 中 s 的前驱状态上实现.路标计数启发式的设计思想是将状态 s 的目标距离估计为从 s 到达目标还需要实现的路标个数.具体地,对于由动作序列 π 所到达的状态 s ,计算路径 π 经过的各个状态已经实现的所有路标:

$$Accepted(s,\pi) = \begin{cases} \{\psi \in L \mid s \models \psi \wedge \exists(\varphi \rightarrow_x \psi) \in O\}, & \pi = \langle \cdot \rangle \\ Accepted(s_0[\pi'], \pi') \cup \{\psi \in L \mid s \models \psi \wedge \forall(\varphi \rightarrow_x \psi) \in O : \varphi \in Accepted(s_0[\pi'], \pi')\}, & \pi = \pi'; \langle a \rangle \end{cases}$$

以及从 s 到达目标状态需要再次实现的路标集合:

$$ReqAgain(s,\pi) = \{\varphi \in Accepted(s,\pi) \mid s \not\models \varphi \wedge (s_* \models \varphi \vee \exists(\varphi \rightarrow_{gn} \psi) \in O : \psi \notin Accepted(s,\pi))\}.$$

$ReqAccepted(s,\pi) \subseteq Accepted(s,\pi)$,从状态 s 需要再次实现路标 φ 的条件是: φ 为目标条件之一;或者在路标图中, φ 必须在相邻时间步上支持其他尚未实现的路标.例如,想要添加其他路标时, φ 作为动作的前提条件必须成立等.最终,从状态 s 开始,需要实现的路标集合为 $L(s,\pi) = (L \setminus Accepted(s,\pi)) \cup ReqAccepted(s,\pi)$,路标计数启发式对状态 s 的估值为 $h^{LM-counting}(s) = |L(s,\pi)|$.

FF 启发式评估过程采用了“忽略动作删除效果”的放松思想,对于每一个访问的状态 s ,利用类似于图规划的方法提取出一个从状态 s 到达目标的放松规划解,以此计算 s 的启发式估值.

定义 3^[8]. 给定一个规划问题 $T = \langle P, A, I, G, C \rangle$,通过忽略动作删除效果所得到的放松规划问题为

$$T^+ = \langle P, A^+, I, G, C \rangle,$$

其中, $A^+ = \{ \langle pre(a), add(a), \emptyset \rangle \mid \langle pre(a), add(a), del(a) \rangle \in A \}$ 是所有放松动作的集合.

定义 4^[8]. 对于一个规划问题 T ,若放松动作序列 $\pi^+ = \langle a_1^+, a_2^+, \dots, a_n^+ \rangle$ 是放松规划问题 T^+ 的一个规划解,则称动作序列 $\pi = \langle a_1, a_2, \dots, a_n \rangle$ 是原始问题 T 的一个放松规划解.

FF 启发式估值(以下记为 h^{FF})的获取包含两个阶段:在以 s 为初始状态的放松规划问题下,前向链接阶段以代价传递的方式计算从当前状态 s 实现每个命题的代价,同时为每个命题记录一个最优支持动作,即能够添加该命题且实施难度最小的动作;反向提取阶段从目标条件开始,选择各个子目标的最优支持动作,最终所有选出的动作构成放松规划解 $\pi(s)$,具体的计算过程可用以下公式表示:

$$\pi(s) = \bigcup_{p \in G} \pi(p;s),$$

$$\pi(p;s) = \begin{cases} \{\cdot\}, & p \in s \\ \pi(a_p;s), & p \notin s \end{cases} \text{ with } a_p = \arg \min_{p \in add(a)} Cost(\pi(a;s)), \pi(a;s) = \{a\} \cup \left\{ \bigcup_{q \in pre(a)} \pi(q;s) \right\}.$$

将动作的执行代价考虑在内,放松规划解的代价即为状态 s 的 h^{FF} 估值: $h_{FF}(s) = Cost(\pi(s)) = \sum_{a \in \pi(s)} C(a)$.

2 路标引导搜索的基本思想

多启发式搜索的模式下,启发式之间有多种不同的结合方式,其中多队列切换的效果最好,这种方式下,启发式各自独立存储与排列 Open 表队列,搜索时交替使用各个启发式作为评价函数,反复从当前启发式的 Open 表中选择要扩展的状态,对所做的选择完全信任,忽略其他启发式对所选状态估值的好坏.

LAMA 规划系统选用路标计数启发式与 FF 启发式执行多队列搜索.通常对启发式好坏的评价主要表现在状态估值所包含的信息量,基于放松规划解的 FF 启发式给出的是近似最优的代价估值,且能够在多项式时间内计算,从求解能力上看,当只采用带有动作代价的 FF 启发式时,虽然多数情况下所得到的规划解质量较好,但成功求解的问题个数较少,加入路标信息后求解的成功率有显著的提高.可见,LAMA 规划系统中路标的使用弥补了 FF 启发式的求解能力.实际上,路标计数启发式的估值 $h^{LM-counting}$ 将目标距离粗略地估计为还需要实现的路标个数,如果单独使用,其本身指导搜索的能力并不强,在实际目标距离并不同的状态上路标信息很可能相同,特别在路标个数不多的问题中,搜索空间中一部分区域内的状态始终取得相同的估值,此时, $h^{LM-counting}$ 给出的信息量并不充足,无法正确区分各个状态接近目标的程度.当路标计数启发式的指向性不强时,搜索空间的扩展方向无法保证越来越靠近目标,从路标计数启发式的 Open 表中取出的状态节点可能暂时偏离了目标,在接下来的搜索过程中要重新修正目标方向将增加额外的计算量.

从所提供的信息量上看,路标计数启发式对各个状态的区分能力远远不如 FF 启发式,在规划求解的过程中,路标计数启发式通常仅起到辅助作用,引导搜索逐渐到达实现了更多路标的状态,可以看作是搜索空间上的分段启发式,最终求得的规划解必须根据路标图给定的次序陆续实现所有的路标,到达目标状态时, $h^{LM-counting}$ 估值降低为 0.以此为理论依据,本节提出一种新的由路标引导搜索的方法,仍然以启发式的方式应用路标信息,在路标计数启发式的引导下对规划任务进行分解,求解过程结合了增强爬山算法与贪心最好优先算法.从全局范围看,规划算法由初始状态开始展开增强爬山过程,每一步爬山都搜索 $h^{LM-counting}$ 估值更低的状态;局部搜索过程不采用盲目搜索的方式,而是根据 FF 启发式对于目标代价的估值 h^{FF} 执行贪心最好优先算法来搜索爬山路径,同时判断 $h^{LM-counting}$ 的改变,每当遇到 $h^{LM-counting}$ 估值更低的状态时,搜索过程实现了更多的路标,距离目标更近,此时,将当前状态作为新的起点,继续向目标方向执行爬山搜索.基本框架如图 1 所示.

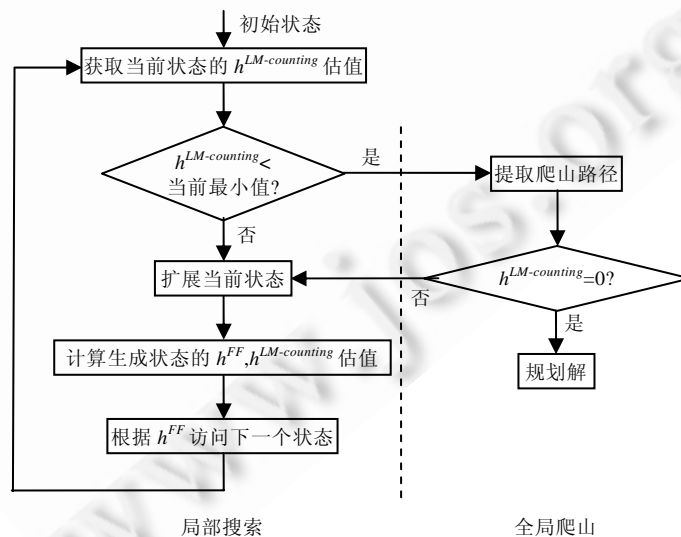


Fig.1 The basic framework of the search procedure

图 1 搜索过程的基本框架

图 1 中:虚线右边部分反复提取爬山路径的过程.可以看出,规划解最终由各段爬山路径链接而成;虚线左边部分是寻找爬山路径的局部搜索过程,对每一个生成状态分别进行路标计数与 FF 两次启发式评估,两个启发式估值分别有不同的用途:1) h^{FF} 估值用于局部搜索过程,根据 h^{FF} 排列 Open 表中的状态节点,选取下一个要访问的状态;2) $h^{LM-counting}$ 用于在全局范围内监督搜索进度,提示爬山过程何时提取爬山路径;如此反复地“搜索-提取”爬山路径即为规划解的生成过程.

3 基于路标计数启发式的增强爬山算法

本节具体给出由路标计数启发式引导的增强爬山算法,需要使用的全局变量包括用于贪心最好优先搜索的 Open 表与 Closed 表, $best_lm_h$ 记录当前已求得的最小值, $current_state$ 跟踪当前的访问状态, $last_state$ 记录上一步爬山所到达的状态,当前局部搜索结束后,通过反向回溯提取从 $last_state$ 到达 $current_state$ 的爬山路径,动作序列 π 用于存储爬山路径.

算法 1. $LM_ehc(\cdot)$.

输入:规划问题 $\langle F, A, I, G, C \rangle$;

输出:规划解或失败退出.

全局变量: $open_list, closed_list,$

$best_lm_h,$

$current_state, last_state,$

π ;

1. 初始化: $open_list \leftarrow \emptyset, closed_list \leftarrow \emptyset,$

$best_lm_count \leftarrow \infty,$

$current_state \leftarrow I, last_state \leftarrow I,$

$\pi \leftarrow \langle \cdot \rangle$;

2. 反复执行步骤 2.1~步骤 2.4,直到 $best_lm_h=0$:

2.1. 若 $current_state \neq last_state$ 且 $current_state \notin closed_list$,则

$closed_list \leftarrow closed_list \cup \{current_state\}$;

获取当前状态的路标计数启发式估值 $h^{LM-counting}$;

2.1.1. 若 $h^{LM-counting} < best_lm_h$,则

$best_lm_h = h^{LM-counting}$;

提取从 $last_state$ 到达 $current_state$ 的动作序列 π' ;

$\pi \leftarrow \langle \pi; \pi' \rangle$;

$last_state \leftarrow current_state$;

$open_list \leftarrow \emptyset, closed_list \leftarrow \emptyset$;

转步骤 2;

2.2. 扩展 $current_state$,对生成的后继状态进行路标计数启发式与 FF 启发式评估,加入 $open_list$;

2.3. 若 $open_list = \emptyset$ 且 $G \not\subseteq current_state$,则失败退出;

2.4. 从 $open_list$ 中选取 h^{FF} 估值最小的状态,指定为当前状态 $current_state$;

3. 返回规划解 π .

算法 1 的输入为给定的规划问题,输出为成功求得的规划解,或算法失败退出.从初始状态开始搜索,步骤 2.1~步骤 2.4 给出的是每一步爬山的局部搜索过程,如果当前状态 $current_state$ 没有扩展过,那么将其加入 Closed 表中.在步骤 2.1.1 中,如果当前所访问状态的 $h^{LM-counting}$ 估值小于当前已求得的最小值 $best_lm_h$,表明更多的路标已被当前状态实现,局部搜索过程找到了爬山过程需要到达的状态,那么从 $current_state$ 回溯到当前局部搜索的起始点 $last_state$ 将获得一段爬山路径 π' ,追加到动作序列 π 之后,至此完成了一次爬山.根据爬山算

法的思想,将 $current_state$ 作为新的起始点,清除先前的搜索信息,继续搜索下一段爬山路径.

如果评估当前状态 $current_state$ 没有得到更小的 $h^{LM-counting}$ 估值,则需要继续向前扩展更多的状态.步骤 2.2 扩展当前状态 $current_state$,算法中省略了具体的扩展过程.为了更准确地使用 FF 启发式,算法对生成的后继状态采用立即估值的策略,在进入 Open 表前就进行启发式评估,根据自身的估值在 Open 表中排序.步骤 2.3、步骤 2.4 选择下一次要访问的状态,到达目标状态之前,如果 Open 表变为空,那么爬山失败;否则,从 Open 表中取出估值最小的状态进行扩展.

下面通过一个简单的积木世界问题描述算法 1 的主要求解过程.

例 1:Block 域规划问题描述的是一类按照指定的目标形态排列积木块的问题,可用动作有:

- $(move-b-to-b ?bm ?bf ?bt)$,表示将积木块 $?bm$ 从积木块 $?bf$ 上移到 $?bt$ 上;
- $(move-b-to-t ?bm ?bf)$ 表示将积木块 $?bm$ 从积木块 $?bf$ 上移到桌子上;
- $(move-t-to-b ?bm ?bt)$ 表示将积木块 $?bm$ 从桌子上移到积木块 $?bt$ 上.

所有动作的执行代价均为统一的单元代价.

图 2 给出的问题实例包括 A, B, C, D 这 4 个积木块和一个桌子 $Table$,具体的问题描述如下:

初始状态: $I=\{(on-table B), (on-table C), (on-table D), (on A B), (clear A), (clear C), (clear D)\}$.

目标条件: $G=\{(on-table B), (on C B), (on D C)\}$.



Fig.2 A planning instance from the Block domain

图 2 Block 域规划实例

虽然目标条件对 A 的位置并没有要求,但整个搜索过程中 A 的位置变化将影响最终规划解的质量.可以看出,为实现目标条件 $(on C B)$,必须先将位于 B 上的积木块 A 移开,然后才能将积木块 C 移到 B 之上.因此,除了初始状态和目标条件给出的命题外,还将另外提取出一个规划执行过程中必须实现的命题路标 $(clear B)$.由于问题的结构简单,例 1 中不包含任何析取命题路标.扩展初始状态时,通过执行动作可以将积木块 A 分别移动到桌子上、 C 上或 D 上,这 3 个后继状态都能实现路标 $(clear B)$.但根据算法 1 给出的步骤,局部搜索过程利用 FF 启发式估值排列所有候选状态,在初始状态的各个后继中,将 A 移到桌子上所得到的状态距离目标最近,因此算法 1 在下一步将访问这个后继状态,实现命题 $(clear B)$,至此完成了一步爬山,提取出爬山路径 $(move-b-to-t A B)$,并得到下一次局部搜索的开始状态为 A, B, C, D 这 4 个积木块都位于桌子上,继续求解时,算法始终能够根据 h^{FF} 估值的排序做出正确的决定,最终求出规划解 $\{(move-b-to-t A B), (move-t-to-b C B), (move-t-to-b D C)\}$.如果在初始状态上选择将 A 移到 C 上或 D 上,虽然也能够实现路标 $(clear B)$,但 A 的位置将妨碍接下来对 C 和 D 的移动,为了得到最终的目标条件,还需要再增加额外的动作将 A 移走,使得规划解的质量有所下降.

由第 2 节中的分析可知,路标计数启发式粗略的估值是多队列切换方法中影响求解效果的主要因素,在同一个搜索过程中,路标计数启发式与 FF 启发式处于同等地位,但这两个启发式的引导能力很难协调.算法 1 给出的规划方法并没有让两个启发式之间信息量的差距成为求解的障碍,反而充分利用了这种差距,在粗略和精确的状态估值之间取得平衡.考虑到路标计数启发式带有“区域性”的特点,令其在全局范围内划分搜索空间,发挥“计数”的功能,克服了在多队列切换方法中引导能力较差的不足,这种处理方式更加符合路标计数启发式的设计思想.同时,由路标引导的增强爬山过程中两个启发式分工明确,从不同的角度发挥引导搜索的能力,主要以 FF 启发式作为前向局部搜索的评判标准,侧重于把握搜索的大方向和对目标代价的估计,在路标计数启发式估值都相等的范围内,每一个访问的状态都由 FF 启发式选出,确保了所选动作都朝目标方向进行;而路标计数启发式在整个求解过程中起到辅助的记录作用,估算各个状态到达目标所需的计算量,侧重于划分每一个局部搜

索过程的执行区域.由此,搜索过程能够综合两个启发式的优势,同时兼顾求解任务的完成情况和目标的实现代价.与多队列启发式搜索方法相比,路标引导的增强爬山能够避免由粗略的 $h^{LM-counting}$ 估值导致的冗余动作.例 1 中,如果使用多队列搜索方法并且当前使用的是路标计数启发式,那么 FF 启发式估值的好坏完全被忽略,将 A 从 B 移动到桌子上、C 上和 D 上所得 3 个状态的 $h^{LM-counting}$ 估值相同,搜索算法无法保证在下一步能够选出距离目标最近的状态.

与文献[12]中给出的方法相比,本文提出的爬山算法没有按照路标信息强制地划分原始任务,虽然算法 1 也是分段求出规划解,但求解之前并不确定每段爬山路径具体实现的是哪些路标,可以看作是对原始任务的隐式分解,随时根据路标的到达情况逐渐追加规划解,引导搜索的方式更加灵活.对于算法的完备性,文献[12]中的任务分解方法存在对可解问题求解失败的情况,这是由于分解后独立的子任务分别针对各自需要实现的路标进行求解,当其中一个子任务失败时,对原始任务的求解也失败.相反地,由路标计数启发式引导的增强爬山算法并没有将路标作为子目标,而是作为搜索进度,整个求解过程中目标不变,根据 FF 启发式选择状态进行访问,能够有效地避免到达“死点”状态.

搜索算法的复杂度可由 Open 表的规模来衡量,对比多队列切换的方式,上述算法描述中只使用一个 Open 表,根据 FF 启发式排列局部搜索过程中所有待扩展的状态,而对于路标计数启发式,不存储对各个状态所求得的估值,仅设置一个全局变量 *best_lm_h* 记录当前最小值.爬山方法采用急切搜索的思想,对所做的决定完全信任,每当完成一步爬山时,不保存先前的搜索信息,将 Open 表与 Closed 表彻底清空,从新的起点开始搜索下一段爬山路径,因此,整个求解过程中 Open 表始终保持合理的规模,进一步控制搜索空间的大小,产生了较好的“聚焦”效果.由路标计数的方式引导增强爬山过程在计算代价方面改善了规划算法的性能,通过保持 Open 表的合理规模,避免搜索空间的膨胀,有效地控制访问状态的数目,减少搜索过程的计算量,进而提高求解效率.

从规划算法的求解框架来看,路标计数启发式的使用也可以看作是一种用于辅助搜索流程的控制策略,路标计数启发式估值相同的局部区域内并不局限于采用 FF 启发式引导贪心最好优先搜索策略,任何一种前向搜索的规划框架以及启发式引导搜索的模式都可以用于搜索爬山路径,实现规划任务的分段求解.同一个求解过程中融合多种搜索框架与启发式,是当前规划研究的热点,加入路标信息的控制后,不同的求解模式对规划算法性能的影响,也是我们未来工作中将要研究的问题.

4 实验测试与分析

本节实现了基于路标计数启发式的增强爬山搜索算法(记为 LM-ehc),对比了其于 LAMA 规划系统(<http://www.plg.inf.uc3m.es/ipc2011-deterministic/ParticipatingPlanners>)的多启发式队列方法(multi-queue)在运用路标计数启发式方面的表现.此外,我们还实现了文献[12]中根据路标排序分解原始任务的方法(记为 LM-d),与本文提出的 LM-ehc 方法进行比较,分析了两者在分段求解方面的能力.为了与 LM-ehc 方法一致,LM-d 方法对于每个子任务的求解也采用 FF 启发式引导的贪心最好优先搜索策略.

本节中 3 种方法的搜索过程统一采用立即估值方式,以便于实验结果的比较,计算 FF 启发式估值时均考虑动作代价,给出对目标代价的估计.选用近两届国际规划比赛上有代表性的规划问题域(<http://ipc.informatik.uni-freiburg.de/Domains>,<http://www.plg.inf.uc3m.es/ipc2011-deterministic/DomainsSequential>)进行测试,这些域中的路标信息具有不同的特点,动作均带有非负的执行代价,指定规划解的总代价为需要优化的目标函数.

测试时只执行 Anytime 机制下的第 1 次搜索过程,主要从求解效率与评估状态的个数方面比较 LM-ehc, Multi-queue 和 LM-d 这 3 种方法生成初始规划解的能力.用于实验测试的机器配置如下:操作系统 Ubuntu 8.0, 处理器 1.90GHz,内存 1.0GB.以下各表中,*T* 表示搜索时间(单位:s),*S* 为求解过程所访问的状态个数,即计算启发式估值的次数,*L/C* 为规划解的长度/代价,“-”表示在给定的时间内无法成功求解(求解时间限定为 1 800s),F 表示求解失败.由于 3 种方法预处理阶段的计算时间均相同,表格中仅给出搜索时间的对比.

4.1 Cybersec 域

Cybersec 域建模一类计算机网络安全问题,域中的可用动作作为计算机运行过程中的各种指令,动作的代价

表示计算机执行该指令可能面临威胁的程度.各个规划任务的初始状态与目标条件均相同,差别在于各个问题实例对同一个指令分别赋予不同的代价,模拟计算机受到威胁时所有可能发生的情况,实验数据见表 1.

Table 1 Results of the Cybersec domain

表 1 Cybersec 域实验对比结果

	LM-ehc			Multi-queue			LM-d		
	T(s)	S	L/C	T(s)	S	L/C	T(s)	S	L/C
p1	0.12	65	43/100	0.28	169	55/16	0.08	61	39/100
p2	0.11	64	43/10	0.19	92	39/10	0.08	61	39/10
p3	0.16	82	45/6	0.18	98	39/6	0.12	75	43/6
p4	0.14	65	43/100	0.18	91	39/100	0.11	65	43/100
p5	0.12	63	43/100	0.21	98	43/100	0.12	67	43/100
p6	0.04	64	38/7	0.07	103	38/7	0.04	74	40/7
p7	0.06	82	42/96	0.07	103	38/96	0.05	78	38/96
p8	0.02	45	32/100	0.04	108	43/12	0.02	71	42/12
p9	0.01	38	31/99	0.05	52	31/99	0.02	48	31/99
p10	0.02	62	31/3	0.02	71	27/3	0.02	53	27/3
p11	0.01	58	34/3	0.03	106	32/4	0.02	80	36/4
p12	0.12	168	60/92	15.59	25 454	56/3	-	-	-
p13	0.1	136	62/184	0.17	203	62/184	0.09	186	60/184
p14	0.09	128	64/185	0.21	314	67/185	0.11	254	64/185
p15	0.11	65	43/7	0.15	81	39/7	0.08	60	39/7
p16	0.16	93	54/214	0.25	136	51/211	0.16	104	54/214
p17	0.13	65	39/139	0.24	109	43/143	0.45	288	39/139
p18	0.14	78	42/102	0.34	180	38/98	0.17	103	44/104
p19	0.11	62	39/1039	0.42	109	43/1043	0.4	297	39/1039
p20	0.08	47	39/1039	0.18	86	39/1039	0.46	288	39/1039
p21	0.04	67	36/106	0.11	118	36/106	0.06	79	38/108
p22	0.05	69	36/996	0.11	114	36/996	0.05	65	36/996
p23	0.05	83	43/163	0.05	116	43/163	0.03	81	46/166
p24	0.02	38	31/1021	0.06	53	31/1021	0.06	290	31/1021
p25	0.02	59	31/61	0.04	111	31/61	0.02	51	27/57
p26	0.02	58	32/72	0.05	103	32/72	0.02	88	38/78
p27	0.24	377	60/980	26.67	42 263	55/85	-	-	-
p28	0.13	154	60/1900	0.22	208	60/1900	6.95	14 962	60/1900
p29	0.11	179	62/1912	0.33	294	62/1912	0.09	137	62/1912
p30	0.11	61	39/109	0.2	97	39/109	1.43	1 084	39/109

由表 1 实验结果可以看出,对于该域中所有的规划任务,LM-ehc 方法都能在较短时间内给出初始规划解,Multi-queue 方法仅在两个问题 p12 和 p27 上速度较慢,而 LM-d 方法在这两个问题上求解失败.实验中发现,Cybersec 域规划任务通常不包含初始状态与目标条件以外的中间路标,从初始状态开始,LM-d 识别出下一步需要实现的就是原始问题的目标条件,因此不对原始问题进行分解,而本文的 LM-ehc 方法始终针对目标计算启发式估值,仍根据目标条件的排序分段求解,实现一个目标条件后,再开展新的局部搜索去搜索其余目标条件.在未实现任何目标条件之前,Multi-queue 方法各个状态的 $h^{LM-counting}$ 估值均相同,这种情况下,路标计数启发式不能够起到正确的引导作用,反而增加了不必要的状态访问.从访问的状态数上可以看出:在所有任务上,LM-ehc 方法访问的状态个数都远远小于 Multi-queue 方法,带来了时间效率上的改善;LM-d 在多数问题上访问的状态数也较少,但性能表现不一致,对于一些易于求解的问题也需要访问大量的状态节点.在规划解质量方面,3 种方法的表现基本一致,对于难度较大的两个问题 p12 和 p27,Multi-queue 方法在评估了大量状态后才能找到一个规划解,但规划解的质量很好;而 LM-ehc 方法对于这两个问题求得的规划解代价较高,但仅需访问少量的状态,显示了路标信息快速引导搜索的能力.由于域中包含的动作较多,随着搜索过程的进行,LM-d 的搜索空间迅速膨胀,对庞大的 Open 表实施操作将耗费较多的求解时间,使得 LM-d 方法对于 p12 和 p27 这两个较难的问题无法在限定的时间内成功求解.

表 2 中分别给出了在 0.1s 与 0.5s 内,LM-ehc,Multi-queue 与 LM-d 方法能够求解问题的个数.从统计数据上可以看出,在生成初始规划解上,LM-ehc 方法对于所有问题实例都能够在 0.5s 内完成求解,更侧重于牺牲规划解质量以换取求解速度,为逐步提高规划解质量的后续迭代过程预留出充足的时间.

Table 2 The ability of generating plans of the Cybersec domain

表 2 Cybersec 域规划解生成能力

<i>T</i> (s)	LM-ehc	Multi-queue	LM-d
0.1	15	10	17
0.5	30	28	26

4.2 Elevators域

Elevators 域模拟的是一类用电梯运送乘客的问题.电梯分为快速电梯和慢速电梯,分别带有不同的运行代价.乘客离开/进入电梯的动作执行代价为 0.每部电梯都有自身可容纳的人数限制和不同的运行范围,乘客在必要的时候可以换乘不同的电梯,目标是将各个乘客分别运送到目的楼层.实验数据见表 3.

Table 3 Results of the Elevators domain

表 3 Elevators 域实验对比

	LM-ehc			Multi-queue			LM-d		
	<i>T</i> (s)	<i>S</i>	<i>L/C</i>	<i>T</i> (s)	<i>S</i>	<i>L/C</i>	<i>T</i> (s)	<i>S</i>	<i>L/C</i>
p1	0.01	45	25/83	0.01	59	26/88	0.05	187	60/137
p2	0.01	83	33/99	0.02	49	24/70	0.03	216	81/214
p3	0.01	38	23/78	0.05	113	38/84	0.33	1 140	153/246
p4	0.02	96	49/163	0.06	116	40/143	0.22	871	206/305
p5	0.03	88	40/107	0.06	109	36/113	0.39	1 262	166/205
p6	0.04	135	57/143	0.09	147	50/142	1.93	3 451	207/256
p7	0.08	230	72/190	0.08	211	61/155	2.06	5 226	401/287
p8	0.07	190	64/152	0.1	217	57/172	1.21	3 127	331/299
p9	0.06	152	65/148	0.11	148	49/158	2.47	5 537	320/297
p10	0.31	654	136/237	0.25	468	89/233	83.13	147 214	351/386
p11	0.04	94	40/123	0.04	79	35/115	1.34	3 366	448/504
p12	0.07	124	63/189	0.06	110	59/196	0.56	975	218/414
p13	0.1	134	63/189	0.31	382	72/248	8.91	11 780	549/671
p14	0.42	498	111/321	0.96	898	106/289	9.12	9 720	941/765
p15	0.64	551	132/311	0.8	628	108/289	11.75	10 632	892/809
p16	0.49	401	110/314	0.73	524	98/290	370.34	284 747	3138/946
p17	1.11	779	144/468	2.08	1 243	120/374	59.38	39 617	2007/1075
p18	2.1	1 177	192/453	-	-	-	-	-	-
p19	2.62	1 413	256/502	14.53	6 263	193/546	-	-	-
p20	1.98	993	140/425	4.28	1 695	163/453	-	-	-
p21	0.37	217	91/266	1.6	853	86/268	7.56	4 858	566/984
p22	5.41	2 022	155/464	8.34	2 698	141/480	270.72	101 097	2384/1618
p23	1.53	472	140/379	3.46	916	120/401	1789.54	557 150	9530/1947
p24	3.73	987	185/485	140.54	28 974	187/628	-	-	-
p25	382.55	74 625	252/603	-	-	-	-	-	-
p26	164.55	26 966	266/745	184.01	25 839	252/707	-	-	-
p27	36.89	5 624	326/755	112.59	13 732	330/769	-	-	-
p28	36.93	5 079	401/834	361.33	40 520	303/848	-	-	-
p29	-	-	-	427.29	42 978	304/968	-	-	-
p30	19.72	2 258	349/640	-	-	-	-	-	-

路标信息给出了电梯需要先后到达的楼层以及乘客换成电梯的次序等信息.由表 3 所示可知:LM-ehc 方法求解此类问题的成功率最高;Multi-queue 方法无法发挥路标对于搜索方向的控制能力,尤其是对于规模较大的问题,路标计数启发式与 FF 启发式交替使用,导致搜索方向容易分散;而 LM-d 方法的求解能力最差,需要访问搜索空间中的大量状态后才能找到规划解,问题规模增大时,更是丧失求解能力.分析可知,LM-ehc 方法在 $h^{LM-counting}$ 估值的引导下搜索方向更加集中,能够快速地穿越搜索空间,耗费的时间更少.从规划解代价上看,与 Multi-queue 方法相比,在个别问题上,LM-ehc 方法求得规划解的质量有所降低,但下降幅度不大,整体上两种方法在规划解质量方面的表现一致.很多问题上,LM-d 方法所求得规划解的长度远远超出代价,可知规划解中包含大量零代价的冗余动作,使得 LM-d 方法的规划解质量最差.图 3 给出了在不同时间内 3 种方法求解问题个数的对比结果.如图 3 所示,LM-ehc 方法生成初始规划解的速度更快,在不同的时间限制内始终能够求解更多的问题.

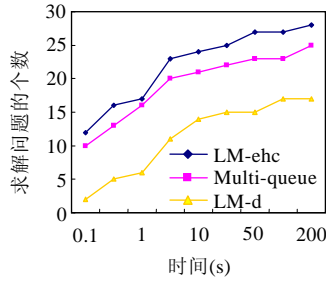


Fig.3 The ability of generating plans of the Elevators domain
图 3 Elevators 域规划解的生成能力

4.3 Transport域

Transport 域描述的是一系列运输问题,每个车辆根据自身的载重限制运输包裹,目标是将各个包裹分别派送到目的地,根据各个车辆行驶路程的长度和装载/卸载的次数计算运输的总花费.实验数据见表 4.

Table 4 Results of the Transport domain
表 4 Transport 域实验对比

	LM-ehc			Multi-queue			LM-d		
	T (s)	S	L/C	T (s)	S	L/C	T (s)	S	L/C
p1	0.00	7	6/54	0.00	8	7/72	<0.01	15	7/72
p2	0.02	30	25/425	0.02	46	26/417	0.02	173	22/344
p3	0.05	66	39/557	0.07	106	36/545	1.62	2 581	36/528
p4	0.19	130	60/878	0.39	252	57/761	4.37	3 771	44/552
p5	0.52	189	70/919	0.66	228	69/845	1.54	661	68/821
p6	3.99	527	174/2151	9.36	1 350	132/1725	55.62	8 757	113/1342
p7	6.39	648	127/1716	7.02	726	104/1228	339.56	34 507	96/1098
p8	18.22	1 183	211/2653	37.45	2 846	118/1218	373.1	28 437	137/1506
p9	17.39	1 008	138/1515	17.97	1 035	122/1353	397.44	22 644	133/1236
p10	34.51	1 620	178/2051	39.19	1 739	154/1539	1089.22	47 782	174/1719
p11	0	12	11/473	0.00	12	11/473	0.01	32	11/473
p12	0.02	89	40/1280	0.08	132	46/1037	0.02	119	35/972
p13	0.05	73	56/1272	0.09	102	55/1337	0.33	510	54/1232
p14	0.62	306	119/2844	0.82	438	94/2165	2.75	1 621	101/2742
p15	1.65	416	186/4940	10.07	2 724	128/2954	8.39	2 568	136/3800
p16	10.95	1 297	269/7043	52.25	7 063	199/4985	119.99	14 133	213/5965
p17	10.42	940	237/6292	38.92	3 328	194/4377	98.43	7 754	253/5540
p18	27.75	1 494	352/7048	83.61	5 430	223/4179	262.04	16 098	393/7740
p19	794.12	39 468	442/10516	792.75	39 739	354/7040	876.92	44 942	477/8938
p20	490.13	20 321	610/13042	1060.35	36 345	376/7387	-	-	-
p21	<0.01	9	8/431	<0.01	9	8/431	<0.01	21	8/431
p22	0.01	22	20/778	0.01	24	20/778	0.04	149	20/778
p23	0.03	46	33/909	0.04	53	31/873	1.68	2 328	30/842
p24	1.39	754	96/2504	0.41	235	65/1740	-	-	-
p25	0.94	229	84/2261	1.74	444	73/2023	-	-	-
p26	7.8	1 017	141/4287	23.98	2 245	108/3088	114.65	12 500	118/2692
p27	49.83	3 987	233/4993	34.56	2 261	143/3577	219.63	16 643	185/4510
p28	30.4	1 563	212/4578	37.93	1 803	128/3186	-	-	-
p29	80.88	3 534	226/4763	76.08	2 877	156/3315	-	-	-
p30	63.64	2 005	335/6900	111.26	3 276	220/5513	-	-	-

该域中提取出的路标信息主要为析取路标形式,指定规划求解过程中各个车辆和包裹必须出现在哪些位置.分析表 4 的测试结果可知,对比 Multi-queue 方法,LM-ehc 在该域上的性能表现不一致,个别问题上路标引导的增强爬山过程并没有加快规划解的生成,求解过程甚至需要访问更大的搜索空间,耗费的时间更长.然而大部分问题上,LM-ehc 方法的时间效率仍有成倍的提升,增强爬山算法在 $h^{LM-counting}$ 的引导下快速取得成功.LM-d 的搜索范围远远超出 LM-ehc,其访问的状态数是 LM-ehc 的几十倍,导致 LM-d 的效率急剧下降.图 4 给出了在不同时间内两种方法求解问题个数的对比,由对比结果可见,整体上 LM-ehc 方法仍显示了更强的求解能力,随着

时间限制的增加,求解问题的个数逐渐超越其余两种方法.

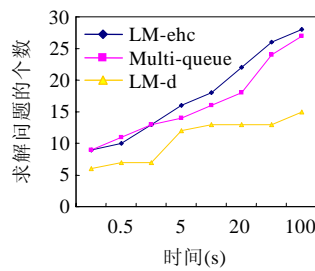


Fig.4 The ability of generating plans of the Transport domain
图 4 Transport 域规划解的生成能力

4.4 Woodworking域

Woodworking 域给出一系列木材加工问题,对木材进行打磨、上色等加工处理,各种加工过程分别使用不同的工具来完成.这类问题的花费为使用不同的工具对木材完成各种加工处理的总代价,实验数据见表 5.

Table 5 Results of the Woodworking domain
表 5 Woodworking 域实验对比

	LM-ehc			Multi-queue			LM-d		
	T (s)	S	L/C	T (s)	S	L/C	T (s)	S	L/C
p1	0.00	8	7/125	0.00	8	7/125	0.01	55	6/120
p2	0.01	28	18/315	0.01	18	17/305	0.01	29	18/325
p3	0.05, 0.05	104	38/930	5.78	4 985	36/895	F	F	F
p4	0.2	258	60/1085	16.13	9 725	55/1030	F	F	F
p5	0.14	114	67/1115	0.14	106	59/1035	F	F	F
p6	2.01	1 586	46/1015	34.03	18 166	49/1065	F	F	F
p7	0.59	220	75/1585	0.73	261	69/1495	F	F	F
p8	14.38	5 748	102/2425	51.15	11 973	81/1960	F	F	F
p9	1.42	334	86/1715	1.12	300	77/1585	F	F	F
p10	5.82	1 769	101/1950	101.01	18 785	106/2050	F	F	F
p11	0.00	6	5/50	0.01	6	5/50	<0.01	8	6/70
p12	0.01	35	19/430	0.01	21	17/390	0.02	63	20/425
p13	0.02	36	31/690	0.03	40	30/660	F	F	F
p14	0.03	39	30/650	0.04	42	30/650	F	F	F
p15	0.11	99	45/1040	0.15	110	46/1055	F	F	F
p16	0.56	384	100/1925	1.59	691	62/1195	F	F	F
p17	0.59	294	115/1735	0.35	152	64/1280	F	F	F
p18	9.43	3 313	86/1790	58.85	16 594	66/1375	F	F	F
p19	2.16	821	145/2655	57.41	12 759	80/1580	F	F	F
p20	1.77	430	104/2075	104.72	16 418	103/2075	F	F	F
p21	0.01	10	9/195	0.01	10	9/195	0.02	69	9/195
p22	0.01	14	13/255	0.01	14	13/255	0.01	22	15/300
p23	0.1	244	25/485	4.35	6 140	26/505	0.84	2 371	26/495
p24	0.29	381	61/1210	44.36	45 087	47/1060	F	F	F
p25	0.63	543	107/1975	22.99	13 029	57/1095	F	F	F
p26	0.67	426	99/1605	33.52	11 568	81/1390	F	F	F
p27	0.45	237	63/1230	28.33	13 040	53/1060	F	F	F
p28	1.35	500	131/2685	0.57	194	84/1835	F	F	F
p29	23.56	7 259	142/2415	139.34	40 579	92/1680	F	F	F
p30	1.81	432	116/2420	2.96	714	97/1990	F	F	F

该域所提取出的路标信息指定了各个木材在加工完成之前必须经过的中间状态,其中大部分为单个命题的形式,仅有少量的析取路标.从表 5 所示的时间效率上看,LM-ehc 方法在所有问题上都显示了较强的优越性,随着问题规模的扩大,LM-ehc 方法对搜索空间的压缩能力更加显著.可以看出:两种启发式交替使用扰乱了搜索方向,导致 Multi-queue 方法并不适用于此类问题;而路标信息能够更准确地挖掘领域知识,根据问题特征进行分段求解.在规划解质量方面,LM-ehc 方法求得的初始规划解质量不如 Multi-queue 方法,但从整体性能上看,

LM-ehc 方法牺牲了规划解的质量来换取时间效率的提高,更符合 Anytime 搜索机制的思想.

由表 5 可以看出,LM-d 方法只能求解几个规模较小的任务,观察实验过程可知,造成这种现象的原因是 Woodworking 域的路标之间通常是紧密相关的,不应该拆分成单独的命题路标.例如,规划任务要得到一个白色的松木块 $p1$,如果 LM-d 方法在求解时将一个柚木制作成 $p1$ 并且处理成白色,虽然也实现了 $p1$ 为白色这一路标,但却无法再实现 $p1$ 为松木这一目标,使得原始任务陷入“死点”,导致求解过程失败.本文提出的 LM-ehc 方法能够避免这种情况的发生,由于爬山过程中只是使用路标信息作为辅助,始终针对目标条件计算 FF 启发式估值来控制状态的访问,因此,LM-ehc 方法下搜索过程不会选取任何“死点”状态.

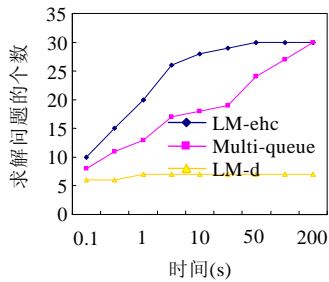


Fig.5 The ability of generating plans of the Woodworking domain

图 5 Woodworking 域规划解的生成能力

的衔接也更加流畅,从而能够控制规划解的质量.

5 结 语

目前,基于路标的启发式被广泛应用于前沿规划系统中,一直受到研究者的关注,路标及其排序关系有效地反映了规划解空间的基本形态,能够帮助规划算法找到规划解.本文研究如何更有效地使用路标信息的问题,根据路标计数启发式与 FF 启发式包含信息度的不同,提出使用两种启发式配合引导搜索的方法.全局范围上,以路标计数启发式作为评价函数,执行增强爬山搜索,爬山过程中的局部搜索执行 FF 启发式引导的贪心最好优先算法.实验结果表明:本文提出的规划求解方法能够更灵活地发挥路标信息的优秀特性,整个搜索过程的执行方向更集中,大幅度地压缩了搜索空间,在求解速度与规划解质量之间取得了较好的权衡,可更快地给出 Anytime 搜索机制下的初始规划解.

研究发现,单个命题形式的路标信息能够更准确地描述规划任务的特征,当析取路标的数量较多时,搜索方向容易出现偏差.例如,当要实现的析取路标为 $L_1 \vee L_2$ 时,对于 L_1 和 L_2 是同等对待的,假设已经到达了 L_1 ,虽然析取路标得到了实现,但很可能先到达 L_2 能够带来质量更好的规划解.此外,进一步挖掘路标之间的关联关系能够避免由零散的路标信息导致的错误决定.在未来的工作中,我们将继续研究如何提取更广泛的路标形式以及不同类型的路标信息对搜索性能的影响,提出更为丰富的路标启发式设计策略.

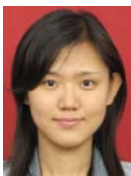
References:

- [1] Bertoli P, Pistore M, Traverso P. Automated composition of Web services via planning in asynchronous domains. *Artificial Intelligence*, 2010,174(3-4):316-361. [doi: 10.1016/j.artint.2009.12.002]
- [2] Gerevini AE, Haslum P, Long D, Saetti A, Dimopoulos Y. Deterministic planning in the 5th Int'l planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence*, 2009,173(5-6):619-668. [doi: 10.1016/j.artint.2008.10.012]
- [3] Blum A, Furst M. Fast planning through planning graph analysis. *Artificial Intelligence*, 1997,90:281-300. [doi: 10.1016/S0004-3702(96)00047-1]

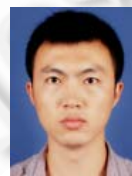
- [4] Bryce D, Cushing W, Kambhampati S. State agnostic planning graphs: Deterministic, non-deterministic, and probabilistic planning. *Artificial Intelligence*, 2011,175(3-4):848–889. [doi:10.1016/j.artint.2010.12.002]
- [5] Wu XJ, Jiang YF, Ling YB. Research and development of StepByStep AI planner. *Ruan Jian Xue Bao/Journal of Software*, 2008,19(9):2243–2264 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/2243.html> [doi: 10.3724/SP.J.1001.2008.02243]
- [6] Lu S, Liu L, Shi L, Li Y. Artificial intelligence planning based on automated reasoning techniques. *Ruan Jian Xue Bao/Journal of Software*, 2009,20(5):1226–1240 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3468.html> [doi: 10.3724/SP.J.1001.2009.03468]
- [7] Bonet B, Geffner H. Planning as heuristic search. *Artificial Intelligence*, 2001,129(1-2):5–33. [doi: 10.1016/S0004-3702(01)00108-4]
- [8] 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]
- [9] Helmert M. The fast downward planning system. *Journal of Artificial Intelligent Research*, 2006,26:194–246. [doi: 10.1613/jair.1705]
- [10] Helmert M, Geffner H. Unifying the causal graph and additive heuristics. In: *Proc. of the 18th Int'l Conf. on Automated Planning and Scheduling*. 2008. 140–147. <http://dblp.uni-trier.de/rec/bibtex/conf/aips/HelmertG08>
- [11] Porteous J, Sebastia L, Hoffmann J. On the extraction, ordering, and usage of landmarks in planning. In: *Proc. of the 6th European Conf. on Planning*. 2001. 37–48. [doi: 10.1.1.28.9990]
- [12] Hoffmann J, Porteous J, Sebastia L. Ordered landmarks in planning. *Journal of Artificial Intelligence Research*, 2004,22:215–278. [doi: 10.1613/jair.1492]
- [13] Richter S, Helmert M, Westphal M. Landmarks revisited. In: *Proc. of the 23rd AAAI Conf. on Artificial Intelligence*. 2008. 975–982. <http://dblp.uni-trier.de/rec/bibtex/conf/aaai/RichterHW08>
- [14] Karpas E, Domshlak C. Cost-Optimal planning with landmarks. In: *Proc. of the 21st Int'l Joint Conf. on Artificial Intelligence*. 2009. 1728–1733. <http://dblp.uni-trier.de/rec/bibtex/conf/ijcai/KarpasD09>
- [15] Bonet B, Helmert M. Strengthening landmark heuristics via hitting sets. In: *Proc. of the 19th European Conf. on Artificial Intelligence*. 2010. 329–334. [doi: 10.3233/978-1-60750-606-5-329]
- [16] Richter S, Westphal M. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 2010,39:127–177. [doi: 10.1613/jair.2972]

附中文参考文献:

- [5] 吴向军,姜云飞,凌应标.智能规划器 StepByStep 的研究与开发. *软件学报*,2008,19(9):2243–2264. <http://www.jos.org.cn/1000-9825/19/2243.html> [doi: 10.3724/SP.J.1001.2008.02243]
- [6] 吕帅,刘磊,石莲,李莹.基于自动推理技术的智能规划方法. *软件学报*,2009,20(5):1226–1240. <http://www.jos.org.cn/1000-9825/3468.html> [doi: 10.3724/SP.J.1001.2009.03468]



魏唯(1984—),女,吉林长春人,博士,讲师,主要研究领域为智能规划,自动推理.
E-mail: wei_wei10@mails.jlu.edu.cn



吕帅(1981—),男,博士,讲师,CCF 高级会员,主要研究领域为智能规划,自动推理.
E-mail: lus@jlu.edu.cn



欧阳丹彤(1968—),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为基于模型的诊断,智能规划,自动推理.
E-mail: ouyd@jlu.edu.cn