

利用分支学习优化子图同构的搜索

张梓涵, 刘燕丽, 李春丽, 迟思义

(武汉科技大学理学院, 湖北武汉 430081)

摘要: 子图同构问题是经典的、具有广泛实际应用的NP完全问题。针对精确算法的分支策略依赖顶点度, 计算代价高的问题, 提出结合无解记录和顶点度约束规则, 通过混合分支学习策略减少求解时间的方法(SIBL)。无解记录是指算法每次重启前无目标解的分支路径, 为了去除无效搜索, 首先移除目标图中顶点度小于当前模式图顶点的候选顶点, 然后移除出现在无解记录中的顶点, 最后依据顶点分值进行降序排序, 优先选择分值大的顶点。新策略提供了利用上界下降量计算单个顶点和顶点匹配的两种分值计算方式, 并交替使用两种分值选择分支顶点以快速寻找目标解, 避免贪心选择的局部最优问题。通过测试14 220个来自生物、图像等领域的算例发现, SIBL相较于当前领先的Glasgow、McSplit+RL_SI分别多解决了10.08%、19.88%的中等难度算例, 验证了分支学习能有效改进子图同构算法的求解效率。

关键词: NP完全问题; 子图同构问题; 分支定界; 约束规则; 分支策略

DOI: 10.11907/rjdk.231223

中图分类号: TP332

文献标识码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1672-7800(2024)003-0088-06



Using Branch Learning to Optimize the Search for Subgraph Isomorphism

ZHANG Zihan, LIU Yanli, LI Chunli, CHI Siyi

(College of Science, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: Subgraph isomorphism problem is a classic and widely applicable NP complete problem. A hybrid branch learning strategy (SIBL) is proposed to reduce the solution time by combining solveless records and vertex degree constraint rules, in order to address the issue of high computational cost and dependence on vertex degree in precise algorithms. No solution record refers to the branch path without a target solution before each restart of the algorithm. In order to remove invalid searches, candidate vertices in the target graph with a vertex degree smaller than the current pattern graph vertex are first removed, and then vertices that appear in the no solution record are removed. Finally, descending sorting is performed based on the vertex score, with priority given to selecting vertices with higher scores. The new strategy provides two score calculation methods that utilize the upper bound descent to calculate a single vertex and vertex matching pairs, and alternately use the two scores to select branch vertices to quickly find the target solution, avoiding the local optimal problem of greedy selection. Through testing 14 220 examples from fields such as biology and imaging, it was found that SIBL is superior to the current leading Glasgow, McSplit+RL_SI solved 10.08% and 19.88% of moderately difficult cases respectively, verifying that branch learning can effectively improve the solving efficiency of subgraph isomorphism algorithms.

Key Words: NP-complete problem; subgraph isomorphism problem; branch-and-bound; constraint rule; branch strategy

0 引言

图是一种非线性结构, 常用于描述事物间的复杂关

系。在给定模式图和目标图中, 子图同构问题(Subgraph Isomorphism, SI)是经典的NP完全问题, 能判断目标图中是否存在子图与模式图同构, 目前已被广泛应用于生物信息^[1]、信息检索^[2-3]、结构检测^[4-5]等领域。图模型强大的

收稿日期: 2023-03-10

基金项目: 国家自然科学基金项目(U22B2017); 湖北省教育厅科学研究计划青年项目(Q20211111)

作者简介: 张梓涵(1996-), 女, 武汉科技大学理学院硕士研究生, 研究方向为组合优化问题求解; 刘燕丽(1980-), 女, 博士, 武汉科技大学理学院副教授, 研究方向为组合优化问题求解、机器学习; 李春丽(1980-), 女, 博士, 武汉科技大学理学院讲师, 研究方向为概率统计; 迟思义(2001-), 女, 武汉科技大学理学院硕士研究生, 研究方向为组合优化问题求解。本文通讯作者: 刘燕丽。

表达能力,使得有关图问题的研究一直受到研究者广泛关注。

早期子图同构精确算法包括回溯算法、动态规划、约束规划等^[6-7]。近年来,研究者们基于回溯算法提出一系列高效的约束规则和分支策略。经典算法VF3的前向检测(Forward-checking, FC)规则通过检查待匹配顶点的邻居顶点是否满足同构条件,以判断当前顶点是否可以匹配^[8]。pathLAD^[9]、Glasgow^[10]算法提出路径的约束规则:如果模式图中任意两个顶点间存在长度为 k 的 m 条路径,那么它们的目标图中映射的像也必定存在相同长度和数量的路径,通过约束规则可去除模式图顶点值域中不满足同构条件的候选顶点,减少计算代价。随着机器学习技术蓬勃发展,研究者尝试将深度强化学习应用于图匹配等相关问题,但由于深度强化学习的训练代价大,目前多用于预处理阶段,难以适用于求解中小规模图^[11]。

然而,上述方法未对搜索过程进行实时学习,无法有效地引导算法优先选择更优路径进行遍历。为此,本文提出一种基于约束规则和分支学习的子图同构算法(Subgraph Isomorphism on Branch Learning, SIBL),利用多种方式去除不满足同构关系的候选顶点,基于分支学习的动态学习方法降低顶点学习的计算代价,适用于求解各种规模图例。

1 相关工作

1.1 相关术语

定义1 设无向、无标签图 $G=(V, E)$, 顶点集 $V' \subseteq V, E' = \{(u, v) \in E | u, v \in V'\}$, 则称 $G[V']=(V', E')$ 是由 V' 诱导的子图。

定义2 给定模式图 $G_p=(V_p, E_p)$ 和目标图 $G_t=(V_t, E_t)$, 如果存在 G_t 的诱导子图 G'_t 和双射 $\varphi: V_p \subseteq V'_t$, 且满足边约束:若 $(\varphi(v), \varphi(v')) \in E'_t$, 当且仅当 $(v, v') \in E_p$, 称 G'_t 与 G_p 为子图同构。

模式图顶点 v 的值域是由满足同构关系、待匹配的目标图顶点集合。图1中 G_p 的顶点1、2、3、4、5构成子图同构, G_t 顶点 c, d, e, f, g 构成子图同构。SI的目标解可表示为匹配对集合,即 $\{(1, f), (3, g), (2, e), (4, c), (5, d)\}$ 。

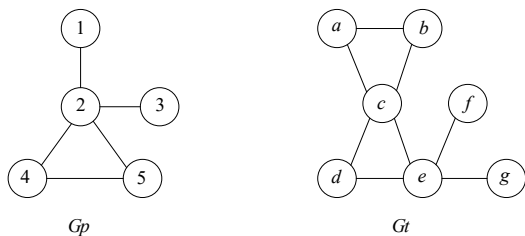


Fig. 1 Example of subgraph isomorphism

图1 子图同构示例

1.2 分支策略

近年来,研究者对子图同构问题的求解提出了多种改进方法,但针对分支策略的改进工作比较有限。最先失败原则(Fail-first rule)的分支策略是优先选择值域小的模式图顶点。从约束规划的角度分析,值域小表示待匹配的尝试次数少,而目标解中必须包含每个模式图顶点的某个匹配,因此先遍历值域小的顶点,有利于简化诱导子图^[12],原则上只需遍历未匹配顶点一次,以寻找值域最小的顶点,因此时间复杂度为 $O(n)$ 。

然而,最先失败原则衍生出多种分支策略变种,例如优先选择所属标签在目标图中出现频率最小的顶点、优先选择单值域的模式图顶点。此外,不少研究者提出基于拓扑关系的顶点选择方式,当算法匹配了顶点 (v, w) , $v \in V_p, w \in V_t$ 后,下一个匹配对优先从 v 和 w 的邻居中选择。优先选择邻域顶点的分支策略保持了顶点之间的相邻关系,有利于目标解的扩展^[12-13]。

2020年,Glasgow^[10,14]算法提出偏好一值排序(Biased value ordering)计算顶点的选择概率,即相同度的顶点具有相同的选择概率,当顶点度每增加1时顶点选择概率增大2倍。不同于前期的贪心策略,偏好一值排序使得度小的顶点也有机会被选为分支点,缓解了因贪心造成的局部顶点被频繁选中的问题。此外,McSplit+RL_SI^[15]算法借鉴强化学习思想,定义了分支顶点的匹配动作和奖励函数,评估顶点在搜索过程中化简子图的作用。该策略实现了对搜索过程的动态学习,有利于算法快速达到搜索树的叶子结点。然而,McSplit+RL_SI仍采用贪心方式,这使得搜索会集中在较少的顶点上。

1.3 同构条件实现

1.3.1 同构条件

同构要求模式图顶点和目标图顶点满足一对一映射,传统算法采用十字链表结构存储顶点值域。初始时,每个模式图顶点的值域是所有目标图顶点,一旦目标图顶点 w 与模式图顶点 v 匹配, w 应从其余顶点值域移除。显然,如此实现同构代价巨大,每次去除冗余的操作时间复杂度为 $O(n^2)$ 。

2017年,McCreesh提出基于域划分的方法,快速实现了同构约束^[16]。具体操作为:设当前分支点为匹配对 (v_i, w_i) ,根据待匹配的模式图(目标图)顶点依据与 $v_i(w_i)$ 是否相邻,将其划分到标签为0或1的两个集合,标签1表示集合内顶点均与分支点相邻,标签0则表示不相邻。具有相同标签的模式图顶点集 V'_p 和目标图顶点集 V'_t 被合并到一个集合,称之为域,记为 $\langle V'_p, V'_t \rangle$ 。若当前分支序列为 $\{(v_1, w_1), (v_2, w_2), \dots, (v_j, w_j)\}$,其中 $v_i \in V_p, w_i \in V_t, 1 \leq i \leq j$,则第 j 层的每个域具有长度为 j 的0-1字符串^[16],字符串的第 k 位表示域中顶点是否与 $v_k(w_k)$ 相邻。

若搜索树的高度为 i ,则当前分支层最多有 2^i 个域。域集是域的集合,记为 DS 。在当前分支,域集可提供的最

大匹配对的个数是沿当前分支,可行解的大小的预估值的上界,如式(1)所示。

$$UB(DS) = \sum_{\langle v_p', v_t' \rangle \in DS} \min(|V_p'|, |V_t'|) \quad (1)$$

式中: $|V_p'|$ 为顶点子集 V_p' 的大小。

1.3.2 域的划分操作与上界计算

如图2所示,给定模式图 G_p 和目标图 G_t ,设分支序列为 $\{(0, a), (1, b)\}$ 。由于顶点2与顶点0相邻,与顶点1不相邻,则字符串为10。未匹配的顶点构成的域集 $DS = \{\langle 2, g \rangle, \langle 3, e \rangle, \langle 4, c, d, f \rangle\}$,域 $\langle 4, c, d, f \rangle$ 的字符串为10。 DS 提供的最大匹配对个数为 $1+1+1=3$ 。

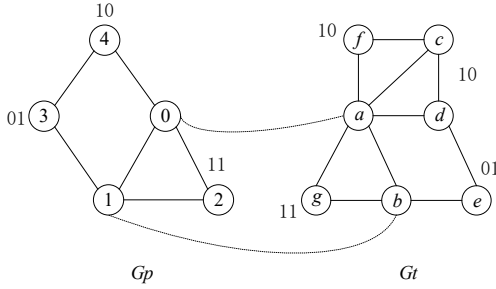


Fig. 2 Example of domain and domain set
图2 域与域集的示例

综上,基于域划分实现同构的时间复杂度为 $O(n)$,因此目前处于先进水平的Glasgow、McSplit+RL_SI和本文提出的新算法均采用此方法实现同构。

2 基于分支学习的子图同构算法

2.1 搜索算法

本文结合顶点度规则和无解记录方法,提出基于分支学习的子图同构算法(Subgraph Isomorphism on Branch Learning, SIBL)以有效剪枝,降低计算代价。算法1描述了SIBL的搜索程序。

算法1 Search($DS, curSol, maxSol, \pi_v, \pi_w$)

输入:域集 DS ,当前解 $curSol$,迄今为止找到的最好解 $maxSol$,模式图(目标图)的顶点选择策略 $\pi_v(\pi_w)$ 。

输出:搜索状态。

1. If $|curSol| > |maxSol|$
2. $maxSol \leftarrow curSol$;
3. If $|curSol| == matchGoal$
4. Return true;
5. $UB = |curSol| + \sum_{\langle v_p', v_t' \rangle \in DS} \min(|V_p'|, |V_t'|)$ //计算上界
6. If $UB \leq |maxSol|$ or $UB \leq matchGoal$ //剪枝
7. If $(--maxIter) == 0$
8. Return restart;
9. Return done;
10. $D \leftarrow selectD(DS)$; //选择某个域
11. $v \leftarrow selectV(V_p, \pi_v)$;
12. $PW \leftarrow possibleW(v, V_t, \pi_w, curSol)$; //筛选候选点
13. For PW 中的每个 w

14. $curSol \leftarrow curSol \cup \{(v, w)\}$; //更新当前解
15. $newDS \leftarrow Partition(DS, v, w)$; //产生新的域集
16. $searchState \leftarrow Search(newDS, curSol, maxSol, \pi_v, \pi_w)$ //递归遍历
17. $curSol \leftarrow curSol \setminus \{(v, w)\}$;
18. If $searchState$ 为true
19. Return true;
20. Else if $searchState$ 为restart
21. 对每个在 PW 位于 w 之前的顶点 w_j ,添加 $curSol \cup \{(v, w_j)\}$ 到 $records$; //添加无解记录
22. Return restart;
23. $newDS \leftarrow$ 从 DS 中移除 v ;
24. $searchState \leftarrow Search(newDS, curSol, maxSol, \pi_v, \pi_w)$
25. If $searchState$ 为true
26. Return true;
27. Else if $searchState$ 为restart
28. 对每个在 PW 位于 w 之前的顶点 w_j ,添加 $curSol \cup \{(v, w_j)\}$ 到 $records$;
29. Return restart;

基于分支定界的算法1的搜索流程,若当前解优于迄今为止找到的最好解,则更新最好解;如果当前解大小已等于目标解 $(|V_p|)$,算法停止搜索。否则,计算当前分支的上界(第1—5行),如果 UB 小于等于当前最好解或小于等于搜索目标,表示当前子树不存在目标解,算法进行剪枝操作。进一步,若迭代次数减少为0,则算法进入重启(第7—8行);否则进入回溯(第9行),从第10行开始描述算法产生新分支的过程。

算法1首先选择出域,然后选出模式图顶点 v 和目标图顶点 w 进行匹配,剩余顶点依据是否与 (v, w) 邻接而被划入新的域集 $newDS$,从而进入递归遍历。若搜索状态为 $true$,表示已找到子图同构;若状态为 $restart$,则记录已经遍历的无目标解的路径,随后进入重启阶段。完成顶点 v 的所有匹配后,从 DS 中移除 v ,然后递归遍历(第23—29行)新域集 $newDS$ 。直至算法遍历所有匹配可能性或达到给定求解时间后,算法停止搜索返回 $false$,即不存在与模式图同构的子图。

$selectD(\cdot)$ 为选择域函数,采用 max_min 策略,首先取该域模式图和目标图顶点个数中的较小值作为域大小,然后从域集中选择最大的域。本文提升分支定界算法效率的关键是减少分支,而高质量的定界函数、约束规则和分支策略是减小搜索空间的主要途径。后续,本文将从约束规则和分支策略两方面提升剪枝率(改进了算法1第11、12行的函数)。具体地,利用约束规则移除不满足同构条件的候选顶点,并设计分支策略优先选择更有利于简化子图的顶点,达到减少分支数的目标。

2.2 改进分支策略

SIBL算法主要采用顶点度和无解记录两种过滤方法,通过减少域中目标图的顶点来减少分支数。顶点度规则是给定制式图 G_p 、目标图 $G_t, \forall v \in V_p, \varphi(v) \in V_t$,根据子图

同构的定义,易知 $\deg(\varphi(v)) \geq \deg(v)$ 。依据顶点度规则,如果 $\varphi(v)$ 顶点度 $< v$ 顶点度,那么 $\varphi(v)$ 不是 v 的像,可进行移除操作。

此外,SIBL采用 Luby 序列设置每次重启迭代的最大次数,为了避免重启后重复遍历无效路径,在重启前存储已遍历、无效的路径(算法 1 的第 21、28 行),并在新的搜索过程中移除在记录库中出现的顶点匹配^[17]。

图 3 为无解记录的搜索树示意图,省略号代表当前结点为根的子树。假设 $curSol = \{ \langle v_i, w_s \rangle \}$, 算法产生分支 $\langle v, w_j \rangle$ 时达到最大迭代次数,将 j 条无解记录 $\{ \langle v_i, w_s \rangle, \langle v, w_1 \rangle \}, \{ \langle v_i, w_s \rangle, \langle v, w_2 \rangle \}, \dots, \{ \langle v_i, w_s \rangle, \langle v, w_j \rangle \}$ 添加到 $records$ 。

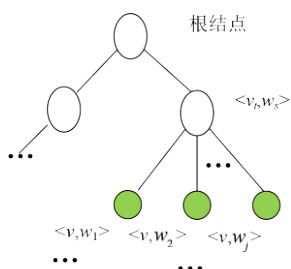


Fig. 3 No good records generation process

图 3 无解记录产生过程

本文优先选择对目标解影响大的顶点进行匹配,算法将更快达到叶子结点。 $newDS$ 的上界作为子图被化简程度的指标,可衡量待解决的子图复杂度,即分支前后上界变化量越大,匹配顶点对问题的求解作用越大。因此,本文提出利用上界的减少量奖励匹配顶点 v, w , 如式(2)所示。

$$R(v, w) = \sum_{\langle v_p', v_i' \rangle \in DS} \min(|V_p'|, |V_i'|) - \sum_{\langle v_p'', v_i'' \rangle \in newDS} \min(|V_p''|, |V_i''|) \quad (2)$$

顶点的价值函数 $q(\cdot)$ 体现了顶点在历史搜索过程中对子图化简的累计作用。式(3)、式(4)分别为单个顶点(模式图或目标图)、匹配对的价值更新公式。其中, $q(v)$ 、 $q((v, w))$ 的初始值为 0。

$$q(v) \leftarrow q(v) + R(v, w) \quad (3)$$

$$q((v, w)) \leftarrow q((v, w)) + R(v, w) \quad (4)$$

由于顶点的单个累计得分综合体现了顶点在历史搜索中的作用,而匹配对的分数体现了在局部环境下顶点的作用,因此设计 SIBL 算法的混合策略,采用交替选择具有最大价值的单个顶点或匹配对进行分支,以平衡探索和利用问题。具体地:策略 π_v 依据式(3)每次从模式图中选择最大分值的顶点 v (算法 1 第 11 行);策略 π_w 依据式(3)、式(4)从目标图中选择最大分值顶点 w 。

算法 2 描述了依据顶点度、无解记录和分支学习,筛选目标图顶点的过程。

算法 2 possibleW ($v, V_p, \pi_w, curSol$)

输入:模式图顶点 v , 目标图顶点子集 V_p , 目标图的顶点选择

策略 π_w , 当前解 $curSol$ 。

输出:模式图顶点排序后的候选集 PW 。

1. PW 初始为空;
2. For 中每个顶点 w
3. If $\deg(v) \leq \deg(w)$
4. If 未在 $records$ 中发现 $curSol \cup \{ \langle v, w \rangle \}$
5. $PW \leftarrow PW \cup \{ w \}$;
6. If policyflag 为 0
7. 依据公式(3)的累计奖励, PW 中的顶点升序排序;
8. Else
9. 依据公式(4)的累计奖励, PW 中的顶点升序排序;
- 10 Return PW ;

由上可知,无解记录采用求解可满足性问题的两文字监督存储结构^[18],判断是否出现在无解记录的操作时间复杂度为 $O(n^2)$,排序函数采用快速排序,时间复杂度为 $O(n \log_2 n)$,因此 possibleW 函数时间复杂度为 $O(n^2)$ 。

3 实验结果与分析

本文实验机器配置为: Intel Xeon CPUs E5-2680V4@2.40 GHz、内存 4 GB。在 Linux 系统中通过 C++ 语言进行编程,每个算例的求解限制时间为 1 000 s。实验比较了 SIBL 与目前处于求解效率领先水平的 McSplit+RL_SI^[15] 和最新 Glasgow 版本^[19]。采用公开、来自实际问题 and 随机模型生成的图同构算例集(14 220 个算例)进行计算^[20]。模式图的顶点数范围为 4~900,目标图的顶点数范围为 10~6 671。

3.1 实验结果

表 1 为各算法在不同算例集上解决的算例个数。实验显示, SIBL 算法相较于 Glasgow (McSplitRL_SI) 多解决了 37(67) 个算例。为了更准确地评估分支学习对困难算例求解的作用,表 1 中算例被分为简单、中等难度。简单算例均被 3 个算法在 10 s 内解决,共计 13 745 个,其余为中等

Table 1 Numbers of solving instances

表 1 各求解算例个数

算例集	McSplit+RL_SI	Glasgow	SIBL
Image	6 278	6 278	6 278
LargerLV	3 310	3 340	3 366
Mesh	3 018	3 018	3 018
PR15	24	24	24
Phase	195	182	198
Scalfree	100	100	100
Si	1 157	1 170	1 165
总计	14 082	14 112	14 149

难度算例。图4比较了各算法在中等难度算例中的求解效率,实验表明SIBL相较于Glasgow(McSplit+RL_SI)多解决了10.08%(19.88%)的算例。在200~1 000 s范围内SIBL解决了更多算例,说明分支学习有利于求解困难算例。

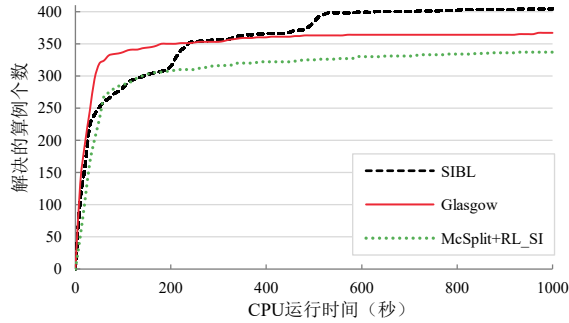


Fig. 4 Number of solutions in moderately difficult examples

图4 中等难度算例中的求解个数

3.2 实验分析

对于存在目标子图与模式图同构的算例,算法1一旦找到目标解即停止搜索(第3、第4行)。因此,目标解越快被找到,搜索树越小。

图5比较了存在目标解的算例搜索树大小,实验数据表明在相同的递归次数下SIBL算法可解决更多算例。此外,各算法的区别仅在于分支策略和筛选候选集方法层面,这说明基于分支学习的混合策略相较于顶点度策略或单个学习策略能更快寻找目标解。

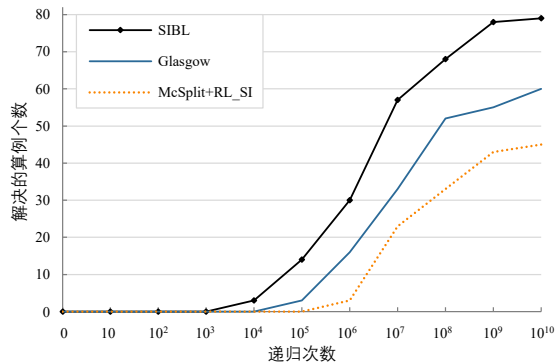


Fig. 5 Search tree size on examples with target solutions

图5 存在目标解的算例上的搜索树大小

对于目标解为false的算例,子图同构算法可分为两个阶段:第一阶段未找到与模式图同构的子图,只找到近似目标解,即maxSol的值接近模式图大小;第二阶段遍历其余搜索空间,证明不存在比maxSol更好的解。

图6、图7的实验结果验证了分支学习策略对这两个阶段的有效性。其中,图6表明SIBL算法相较于McSplit+RL_SI、Glasgow算法在相同运行时间下,找到近似目标解的算例更多,说明分支学习策略有利于在目标图中更快地寻找与模式图同构的最大公共子图。由图7可见,当递归次数大于10⁸,SIBL算法解决了更多算例,说明分支混合策略能更快证明目标解不存在。

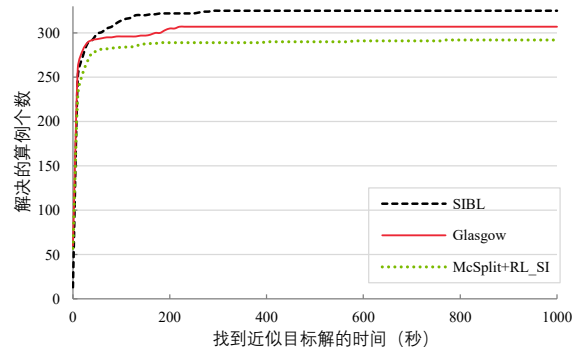


Fig. 6 Time to find approximate target solution in moderately difficult examples

图6 中等难度算例中找到近似目标解的时间

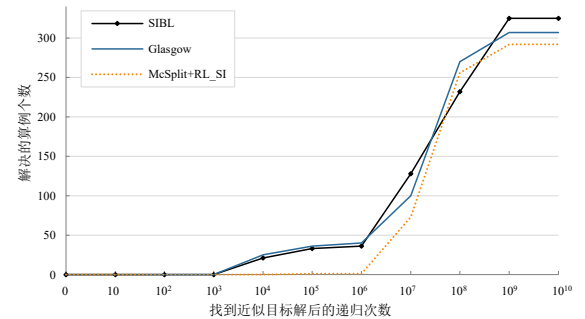


Fig. 7 Number of recursions to prove that there is no objective solution

图7 证明不存在目标解的递归次数

4 结语

子图同构是广泛应用于社交网络、集成电路等领域的经典NP完全问题。为了更快地确定目标图中是否存在子图与模式图同构,本文基于分支定界的SI算法提出新的分支学习策略,该策略改进了传统SI算法依赖顶点度或单一学习策略的不足,在减少搜索冗余的同时实现了顶点的多样化选择,有效提升了分支定界SI算法的效率,证明了评估匹配动作可降低问题产生的影响。

然而,来自实际问题的图大多数属于稀疏图,例如社交网络挖掘,寻找重要的顶点更有利于将原问题分解为若干小规模子问题进行求解,对工程求解具有实际意义。因此,后续研究工作将围绕环境和动作的交互学习展开。

参考文献:

- [1] SEONG W, SIJAO W, JOYCE J, et al. HiddenCPG: large-scale vulnerable clone detection using subgraph isomorphism of code property graphs [C]// ACM Web Conference 2022, 2022:755-766.
- [2] ZHANG W J, CHAN W K. Subgraph isomorphism building on a hierarchical query graph [C]// 5th International Conference on Compute and Data Analysis, 2021:107-111.
- [3] CHEN L, CHEN X S, LUO Y G, et al. Compliance analysis method of Hive data operation based on subgraph isomorphism [J]. Journal of Elec-

- tronics and Information Technology, 2022, 44(12):4367–4375.
- 陈丽,陈兴蜀,罗永刚,等. 基于子图同构的Hive数据操作合规分析方法[J]. 电子与信息学报, 2022, 44(12):4367–4375.
- [4] FENG J Y, ZHANG X W, FENG Z Y. Parallel algorithms for RDF type-isomorphism on GPU [J]. Journal of Computer Research and Development, 2018, 55(3): 651–661.
- 冯佳颖,张小旺,冯志勇. 基于GPU的RDF类型同构并行算法[J]. 计算机研究与发展, 2018, 55(3): 651–661.
- [5] BONNICI V, GIUGNO R, BOMBIERI N. An efficient implementation of a subgraph isomorphism algorithm for GPUs[C]// 2018 IEEE International Conference on Bioinformatics and Biomedicine, 2018: 2674–2681.
- [6] YU J, LIU Y B, ZHANG Y, et al. A survey of large-scale graph data matching techniques [J]. Journal of Computer Research and Development, 2015, 52(2): 391–409.
- 于静,刘燕兵,张宇,等. 大规模图数据匹配技术综述[J]. 计算机研究与发展, 2015, 52(2): 391–409.
- [7] XIANG Y Z, TAN J X, HAN J S, et al. Survey of graph matching algorithms[J], Computer Science, 2018, 45(6): 27–33.
- 项英倬,谭菊仙,韩杰思,等. 图匹配技术研究[J]. 计算机科学, 2018, 45(6): 27–33.
- [8] CARLETTI V, FOGGIA P, SAGGESE A, et al. Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with VF3 [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 804–818.
- [9] KOTTHOFF L, MCCREESH C, SOLNON C. Portfolios of subgraph isomorphism algorithms [C]// Proceedings of International Conference on Learning and Intelligent Optimization, 2016: 107–122.
- [10] MCCREESH C, PROSSER P, TRIMBLE J. The Glasgow subgraph solver: using constraint programming to tackle hard subgraph isomorphism problem variants[C]// Proceedings of International Conference on Graph Transformation, 2020: 316–324.
- [11] BOURITSAS G, FRASCA F, ZAFEIRIOU S, et al. Improving graph neural network expressivity via subgraph isomorphism counting [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(1): 657–668.
- [12] BONNICI V, GIUGNO R. On the variable ordering in subgraph isomorphism algorithms[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2016, 14(1): 193–203.
- [13] XU Z P, LI Z, LIU H D, et al. Subgraph isomorphism matching algorithm based on neighbor information aggregation[J]. Journal of Computer Applications, 2021, 41(1): 43–47.
- 徐周波,李珍,刘华东,等. 基于邻居信息聚合的子图同构匹配算法[J]. 计算机应用, 2021, 41(1): 43–47.
- [14] ARCHIBALD B, DUNLOP F, HOFFMANN R, et al. Sequential and parallel solution-biased search for subgraph algorithms[C]// Proceedings of the International Conference on Integration of Constraint Programming, Artificial Intelligence and Operations Research, 2019: 20–38.
- [15] LIU Y, LI C M, JIANG H, et al. A learning based branch and bound for maximum common subgraph related problems [C]// Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020: 2392–2399.
- [16] MCCREESH C, PROSSER P, TRIMBLE J. A partitioning algorithm for maximum common subgraph problems [C]// Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017: 712–719.
- [17] LUBY M, SINCLAIR A, ZUCKERMAN D. Optimal speedup of Las Vegas algorithms [J]. Symposium Theory and Computing Systems, 1993, 47(4): 128–133.
- [18] LEE J, SCHULTE C, ZHU Z. Increasing nogoods in restart-based search[C]// Thirtieth AAAI Conference on Artificial Intelligence, 2016: 3426–3433.
- [19] SONJA K, MCCREESH C. Solving graph homomorphism and subgraph isomorphism problems faster through clique neighbourhood constraints [C]// Thirtieth International Joint Conference on Artificial Intelligence, 2021: 1396–1402.
- [20] SOLNON C. Benchmarks for the subgraph isomorphism problem [EB/OL]. <http://liris.cnrs.fr/csolnon/SIP.html>.

(责任编辑:刘嘉文)