arXiv:2301.11927v1 [cs.DM] 26 Jan 2023

PACE 求解器描述：启发式有向反馈顶点集问题算法

AndreiArhire ！ 亚历山德鲁·约安·库扎大学，
罗马尼亚雅西

Paul Diac
亚历山德鲁·约安·库扎大学，罗马尼亚雅西

摘要

图的反馈顶点集是一个节点集合，其性质是每个环都至少包含该集合中的一个顶点；也就是说，移除反馈顶点集中的所有顶点后，图将变成无环图。在这篇短文中，我们介绍了 _UAIC_ANDREIARHIRE_ 求解器所使用的最小有向反馈顶点集查找算法。该求解器提交参加了 2022 年 PACE 挑战赛的启发式赛道。

2012 年 ACM 主题分类 计算理论 →随机局部搜索；计算理论 → 图算法分析

关键词与短语 有向反馈顶点集、局部搜索

补充材料 源代码可在 Zenodo（10.5281/zenodo.6646187）和 GitHub（https://github.com/AndreiiArhire/PACE2022）上获取。

1 预备知识

设 $G=(V,E)$ 是一个有向图。

- $N_G^-(v)=\{u\in V\mid \exists(u,v)\in E\wedge \nexists(v,u)\in E\}$, $d_G^-(v)=|N_G^-(v)|$;$N_G^+(v)=\{u\in V\mid \nexists(u,v)\in E\wedge \exists(v,u)\in E\}$, $d_G^+(v)=|N_G^+(v)|$;$N_G^\pm(v)=\{u\in V\mid \exists(u,v)\in E\wedge \exists(v,u)\in E\}$, $d_G^\pm(v)=|N_G^\pm(v)|$;$G-v=(V\backslash\{v\}\ E\cap\{\{V\backslash\{v\}\}\times\{V\backslash\{v\}\}\})$;$G\circ v=(V\backslash\{v\}\ E\cap\{\{\{V\backslash\{v\}\}\times\{V\backslash\{v\}\}\}\cup\{\{N_G^-(v)\cup N_G^\pm(v)\}\times\{N_G^+(v)\cup N_G^\pm(v)\}\}\})$;$G$ 是双向团，当且仅当 $\forall x,y\in V$, $x\neq y$ 且 $\{(x,y),(y,x)\}\subset E$。

我们将 $G-v$ 操作称为顶点 v 删除（移除 v 及其所有邻接边），将 $G\circ v$ 操作称为顶点 v 合并（连接其所有前驱和后继，并将 v 从图中移除）。

2 求解器摘要

我们的求解器所使用的算法包含三个主要阶段。第一阶段会找到一个初始解。第二阶段会移除初始解中包含的冗余顶点。第三阶段则基于已知最优解执行若干次局部搜索。

3 约简规则

该问题可以通过以下方式理解：我们必须将每个顶点放入集合 A 或集合 B，使得最终 A 是最小反馈顶点集，而 B 是无环剩余部分。我们可以根据某些能够在多项式时间内完成的验证，将一组顶点放入 A 或 B。

2 PACE 求解器描述：启发式有向反馈顶点集问题算法

以下两种操作被视为约简：如果某个顶点可以成为集合 A 的一部分而不影响解的最优性，则将其移除，并最终将其引入解集中。如果某个顶点可以成为集合 B 的一部分而不影响解的最优性，则将其合并。

我们使用了 [4], [2] 和 [3]中描述的 8 条约简规则。

▶ **约简规则 1**。 如果存在顶点 $v \in V$和边 $(v, v) \in E$，则移除 v。

在这种情况下，顶点 v 包含一个自环。将其删除并加入解中。

▶ **约简规则 2**。 如果存在顶点 $v \in V$、$(v, v) \notin E$，且满足 $d_G^-(v) + d_G^{\pm}(v) \leq 1$ 或$d_G^+(v) + d_G^{\pm}(v) \leq 1$，则合并 v。

顶点 v 可以有 $d_G^+(v) = 0$ 或 $d_G^-(v) = 0$，此时它不可能属于任何环。否则，将$d_G^+(v) = 1$ 或 $d_G^-(v) = 1$中的所有前驱与所有后继相连，并将v 从图中移除（合并操作），不会影响最优性。在这两种情况下， v 都不会被纳入解中。

▶ **规约规则 3**。 如果存在顶点 $v \in V$，$(v, v) \notin E$，$\min(d_G^-(v)\ d_G^+(v)) = 0$和$N_G^{\pm}(v)$构成一个双向团，则移除 $N_G^{\pm}(v)$并合并 v。

这里的关键观察是，在一个双向团中，至多有一个节点不属于解，因为任意两个顶点都会形成一个环。因此，可以将双向团视为单个节点，从而进一步应用规则 2。

▶ **规约规则 4**。 如果存在顶点 $u, v \in V$，$(u, v) \in E \wedge (v, u) \notin E$，并且 u、 v 和 $\notin E$ 位于图 $(V, E \setminus \{E \cap \{(x, y)\ \forall x \in V, y \in N_G^{\pm}(x)\}\})$的不同强连通分量中，则删除 (u, v)。

不同强连通分量中的顶点之间的边不属于任何环（否则，这些顶点就会属于同一个分量）。因此，可以删除所有这类边。此外，对于含有两个顶点的双向团，至少会移除其中一个节点，因此在计算强连通分量时，可以忽略属于双向团的边。

▶ 约简规则 *5.* 若存在顶点 u、 $v \in V$，$(u$、 $v) \in E \wedge (v$、 $u) \notin E$以及 $(N_G^-(u) \subset \{N_G^-(v) \cup N_G^{\pm}(v)\}) \vee (N_G^+(v) \subset \{N_G^+(u) \cup N_G^{\pm}(u)\})$，则删除 $(u$、 $v)$。

这条规则的思路是删除一组边，使得没有极小环会经过这些边，因为计算反馈顶点集时只需打断极小环。

以下三条约简规则是基于约简规则 3 得出的， r
并结合了这样的思路：双向团中至多只有一个节点不属于解。

▶ **约简规则 6**。 如果存在一个节点 $v \in V, (v, v) \notin E$，使得 $\{N_G^+(v) \cup N_G^{\pm}(v)\}$ 或 $\{N_G^-(v) \cup N_G^{\pm}(v)\}$ 构成一个双向团，则合并 v。

▶ **约简规则 7**。 如果存在一个节点 $v \in V, (v, v) \notin E$，且 $\{N_G^-(v) \cup N_G^+(v) \cup N_G^{\pm}(v)\}$ 可以分成两个集合， $N_G^{\pm}(v)$ 包含在其中一个集合中，且每个集合都构成一个双向团，则合并 v。

▶ **约简规则 8**。 如果存在一个节点 $v \in V$，$(v, v) \notin E$, $d_G^{\pm}(v) = 0$ 以及 $\{N_G^+(v) \cup N_G^-(v)\}$ 可以被划分为至多三个集合，且每个集合都构成一个双向团，则合并 v。

4 第一阶段

在算法的第一部分中，会不断应用约简规则，直到图不再支持任何约简规则。如果图不为空，则选择一个有希望的顶点，将其作为解的一部分并移除。重复执行这两个操作，直到图变为空。若顶点 v 在选择过程中，相较于其他所有顶点，使以下两者之一达到最大值，则认为它是最佳候选：$(d_G^+(v) + d_G^\pm(v)) \cdot (d_G^-(v) + d_G^\pm(v))$ 或 $d_G^\pm(v) \cdot \infty + d_G^-(v) \cdot d_G^+(v)$。

5 第二阶段

经过多次顶点选择后，其中一些顶点可能会在解中变得冗余，即它们不属于任何环，因此可以将其从反馈顶点集中排除。为了最大化排除顶点的数量，我们按照顶点插入顺序的逆序取出这些顶点，并将它们重新引入无环图中。对于某个固定顶点，我们使用用于计算强连通分量的 Tarjan算法优化版本，检查所得图是否为无环图。如果图是无环的，则从解中删除该顶点，并保留图所发生的变化。此步骤在 [3] 和 [1]中进行了介绍。

6 第三阶段

在算法的最后部分，会执行一些局部搜索 [3]。局部搜索包括在 G 的一个子图上运行与第一阶段中所介绍过程相同的流程。子图是通过从目前找到的最佳反馈顶点集中均匀随机移除一个特定子集，并从无环剩余部分中移除另一个特定子集而得到的。此外，根据剩余时间，还会对新解应用第二阶段算法的一个略有修改的版本。

7 复杂度分析

两种选择准则都很简单，但并非总是最优，却能带来显著收益。仅根据相邻邻居的数量进行判断，可利用哈希集合高效执行一些约简，并将大部分数据保留在内存中。每次更新图后，均可在 $\mathcal{O}(1)$ 时间内应用约简 1 和 2。约简 3 的实现复杂度为 $\mathcal{O}((|V| + |E|) \cdot \log(|V|))$，约简 4 的实现复杂度为 $\mathcal{O}(|V| + |E|)$。其余约简仅针对度不超过某个较小常数的顶点运行，我们假设其复杂度为 $\mathcal{O}(|V| + |E|)$。前两种约简在每次图发生变化后运行；其余约简则在边数减少约 5% $-$ 25% 后运行一次。对于 T 次迭代中的每一次，我们都会选择将约 30% 的顶点恢复到无环剩余部分，并运行局部搜索。由于一个实例可以在线性时间内约简为一个具有 $|V| < |E|$ 的图，根据主定理 [5],，程序的最终复杂度为$\mathcal{O}(T \cdot (|V| + |E| \cdot \log|E|))$，其中 T为局部搜索的次数， T 为受时间限制约束的参数。

参考文献

1 Berend Hasselman。《检测冗余反馈顶点的高效方法》。CPB 讨论论文第29号，荷兰经济政策分析局 CPB，2004年4月。网址： https://ideas.repec.org/p/cpb/discus/29.html。2 Hen-Ming Lin；Jing-Yang Jou。《通过收缩操作计算有向图的最小反馈顶点集》。 *IEEE* 集成电路与系统计算机辅助设计汇刊，19：295–307，2000年。网址： http://doi.org/10.1109/43.833199, doi:10.1109/43.833199。3 Mile Lemaic。《有向图反馈顶点集问题的基于马尔可夫链的启发式方法》。博士学位论文，科隆大学，2008年。4 Hanoch Levy；David W Low。《一种用于寻找小型环割集的收缩算法》。 算法杂志，9：470–493，1988年。网址: http://doi.org/10.1016/0196-6774%2888%2990013-2, doi:10.1016/0196-6774(88)90013-2。5 R.M. Verma。《一种通用方法及用于分治递推式的主定理及其应用》。 算法杂志，16：67–79，1994年。网址： http://doi.org/10.1006/jagm.1994.1004， doi:10.1006/jagm.1994.1004。