最小内向森林问题

浙江省杭州学军中学 张哲宇

摘 要

最小内向森林问题是指，对于一张带权的有向图，求固定边数的最小内向森林。本文会围绕最小内向森林问题展开，并给出时间复杂度为 $O(E\log E)$ 的算法。

1 引言

对于一张带权的有向图，并给定一个自然数 k，求最小的共包含 k 条边的子图，满足这个子图中每一个弱连通块都是一棵内向树。这样的问题就是最小内向森林问题，该问题在近几年才首次出现在算法竞赛中，是一个较为新颖的问题。

早在十几年前，国内的评测网站上就出现了最小树形图问题。但是，直到现在，树形图相关的问题依旧是一类冷门问题。最小内向森林问题拥有一定的潜力，可能成为一类用途广泛的图论模型，具有研究的意义。

如今，最小内向森林问题的解决方法通常是基于凸优化和最小树形图的算法，该算法效率低下、功能弱且扩展性差。本文提出了一个时间复杂度为 $O(E\log E)$ 的算法，该算法效率高，功能强且扩展性好。

本文共有七节，每节的内容大致如下。

第二节中，精确地定义了最小内向森林问题。

第三节中，简单介绍了拟阵。拟阵作为组合数学的一个分支，现在常常被用于解决各种图论中的最优化问题。本节中，将会用拟阵来描述最小内向森林问题，并借助拟阵发现性质。

第四节中，给出了基于凸优化和最小树形图的算法。最小树形图问题早在上个世纪六十年代就已经被解决，解决的算法被称为“朱刘算法”或“Edmonds' algorithm”。最小树形图问题与最小内向森林问题同为树形图相关问题，有些许相似之处。本节中，会详细介绍如何使用凸优化将最小内向森林问题转化成最小树形图问题，该算法需要使用二分，二分的次数决定了该算法的精度，当二分次数为 $\log_{\frac{1}{2}}\varepsilon$ 的时候，该算法的时间复杂度为 $O\left(E\log E\log_{\frac{1}{2}}\varepsilon\right)$。

第五节中，分析前几节的内容并得到优先内向树扩张算法。基于凸优化和最小树形图的算法拥有很多冗余操作，剖析这些操作的意义可以使算法更加精简。本节中，会详细

介绍如何得到优先内向树扩张算法，然后形式化地将其描述出来，最后简单说明如何得到 $O(E\log E)$ 的时间复杂度。

第六节中，举了一例最小内向森林问题的应用。

第七节中，对本文内容进行总结。

2 定义

给定一张有向图 $G=(V,E)$，其中 V 是有向图的点集，E 是有向图的边集。有边权 $w:E\mapsto\mathbb{R}$。

定义 **内向树** 为一个弱连通块，没有环，其中有一个根节点，所有的非根节点恰有一条出边。

定义 **内向森林** 为一个有向图，其中的每一个弱连通块都是一棵内向树。

定义 **边子集** 为边集 E 的一个子集。

定义一个边子集 E^* 的 **边导出子图** 为由原图中所有点和 E^* 中的所有边构成的有向图。

规定一个边子集 E^* 的权值等于其中每条边的权值和，即 $w(E^*)=\sum_{e\in E^*}w(e)$。

那么，最小内向森林问题可以被如下描述：

给定一张有向图和一个整数 k，求权值最小的大小为 k 的边子集，满足其边导出子图是内向森林。

对于同一张有向图，记最小内向森林问题的答案关于 k 的函数为 MDF(k)。

3 拟阵

3.1 定义

有一个子集系统 $M=(S,\mathcal{I})$。如果 M 满足以下三个公理，则 M 是一个拟阵。

公理 1. $\emptyset\in\mathcal{I}$。

公理 2. $\forall A\subset B$，若 $B\in\mathcal{I}$，则 $A\in\mathcal{I}$。

公理 3. $\forall|A|<|B|$，若 $A\in\mathcal{I}$ 且 $B\in\mathcal{I}$，则 $\exists x\in B-A$，$A\cup\{x\}\in\mathcal{I}$。

对于拟阵 $M=(S,\mathcal{I})$，称 $\mathcal{I}$ 中的元素为独立集。

3.2 可截取性质

引理 3.1. 给定拟阵 $M=(E,\mathcal{I})$ 和任意自然数 s，那么保留大小不超过 s 的独立集后依旧是拟阵，即 $M^*=(E,\mathcal{I}^*)$ 是拟阵，其中 $\mathcal{I}^*=\{S\in\mathcal{I}:|S|\le s\}$。

证明. 因为 $s \geq 0$，所以空集是独立集，即 $\emptyset \in \mathcal{I}^*$。

$\forall A \subset B$，若 $B \in \mathcal{I}^*$，则 $B \in \mathcal{I}$，所以 $A \in \mathcal{I}$。又因为 $|A| \leq |B| \leq s$，所以 $A \in \mathcal{I}^*$。

$\forall |A| < |B|$，若 $A \in \mathcal{I}^*$ 且 $B \in \mathcal{I}^*$，则 $A \in \mathcal{I}$ 且 $B \in \mathcal{I}$，所以 $\exists x \in B - A$，$A \cup \{x\} \in \mathcal{I}$。又因为 $|A| < |B| \leq s$，则 $|A| + 1 \leq s$，所以 $\exists x \in B - A$，$A \cup \{x\} \in \mathcal{I}^*$。 □

3.3 拟阵与最小内向森林问题

接下来我们使用拟阵表示最小内向森林问题。

令 $M_1 = (E, \mathcal{I}_1)$，其中 E 为边集，$\mathcal{I}_1$ 为边导出子图中把边视为无向边后没有环的边子集的集合。

令 $M_2 = (E, \mathcal{I}_2)$，其中 E 为边集，$\mathcal{I}_2$ 为边导出子图中每个点出度不超过 1 的边子集的集合。

接下来我们分别阐明 M_1 和 M_2 是拟阵，也就是说它们都满足三个公理。

对于 M_1，显然满足 **公理 1** 和 **公理 2**。接下来分析 **公理 3**，$\forall |A| < |B|$，若 $A \in \mathcal{I}_1$ 且 $B \in \mathcal{I}_1$，那么 A 的连通块数量大于 B 的连通块数量，必然存在一条边 $e \in B$，连接 A 中的两个连通块，这便是 $B - A$ 中可以加入到 A 中的元素。所以 **公理 3** 也满足，M_1 是一个拟阵。

对于 M_2，显然满足 **公理 1** 和 **公理 2**。接下来分析 **公理 3**，$\forall |A| < |B|$，若 $A \in \mathcal{I}_2$ 且 $B \in \mathcal{I}_2$，令 A 中有出边的点集为 $U(A)$，B 中有出边的点集为 $U(B)$，则 $|U(A)| \leq |U(B)|$，只需要选择任一以 $U(B) - U(A)$ 中结点为起点的在 B 中的出边，加入到 A 中即可。所以 **公理 3** 也满足，M_2 是一个拟阵。

引理 3.2. 对于任意边集 E^*，$E^* \in \mathcal{I}_1 \cap \mathcal{I}_2$ 当且仅当 E^* 的边导出子图是内向森林。

证明. 先证明，若 $E^* \in \mathcal{I}_1 \cap \mathcal{I}_2$ 可得 E^* 的边导出子图是内向森林。对于 E^* 的边导出子图中的一个弱连通块，由 $\mathcal{I}_1$ 的定义可得，在该弱连通块内，边数比点数少一。由 $\mathcal{I}_2$ 的定义可得，每个点出边数量不超过 1，所以其中恰有一个点没有出边，不妨设这个点为 t。那么，每一条和 t 相连的边都指向 t，以此类推即可得到以 t 为根的内向树。既然 E^* 的边导出子图中的每一个弱连通块都是内向树，所以 E^* 的边导出子图是内向森林。

另一个方向的证明通过内向森林的定义可得。 □

综上，现已使用拟阵来描述最小内向森林问题，即求 $\min\limits_{S \in \mathcal{I}_1 \cap \mathcal{I}_2, |S| = k} w(S)$。

因为最小内向森林问题固定了边的数量，所以所有的边的边权可以同时加减。为了方便下文的讨论，我们不妨设 $\max\limits_{e \in E} w(e) \leq 0$。因此，我们可以将问题变成 $\min\limits_{S \in \mathcal{I}_1 \cap \mathcal{I}_2, |S| \leq k} w(S)$。

3.4 拟阵交和线性规划

3.4.1 拟阵交问题

拟阵交问题的定义：给定两个定义在相同基础集上的拟阵 $M_1 = (E, \mathcal{I}_1), M_2 = (E, \mathcal{I}_2)$，

求 $\min_{S\in\mathcal{I}_1\cap\mathcal{I}_2} w(S)$。

最小内向森林问题可以被描述成一个标准的拟阵交问题。令 $M_2^* = (E, \mathcal{I}_2^*)$，其中 $\mathcal{I}_2^* = \{S \in \mathcal{I}_2 : |S| \le k\}$，$M_2^*$ 是拟阵。那么 $\min_{S\in\mathcal{I}_1\cap\mathcal{I}_2,|S|\le k} w(S) = \min_{S\in\mathcal{I}_1\cap\mathcal{I}_2^*} w(S)$。

3.4.2 线性规划

线性规划问题可以这样描述：给定一个矩阵 A 和两个向量 b, c。

$$\begin{aligned} \min \quad & c^T x. \\ \text{s.t.} \quad & Ax \le b, \\ & x \ge 0. \end{aligned}$$

其中 x 是变量，第一行表示最优化的目标，其余表示对 x 的约束。

如果只关注限制，可以发现 x 的可行范围是一个 $|c|$ 维的凸多胞形。而向量 c 的作用只是规定了最终要求哪一个方向上的极点，即某一个方向上的最远点。

如果，对于任意的 c，最优解中 x 都是整点，即凸多胞形上的任意一个极点都是整点。那么，这样的线性规划称为幺模的。当线性规划是幺模的时候，通常可以无视特定问题中的整数限制。

定理 3.1. *对于一个线性规划和一个向量 d，令 $F(k)$ 表示增加任意一条限制 $d^T x \le k$ 后线性规划的值。则，$F(k)$ 的导数 $F'(k)$ 单调。*

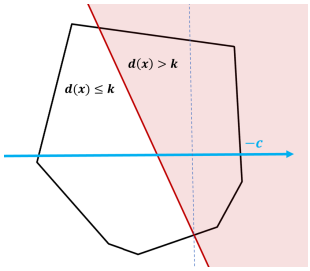


图 1: 图 2.1

证明. *考虑加入限制之前，x 可行范围是一个凸多胞形，将其映射到 c 与 d 所在的平面上后一定是一个二维凸包，如图 2.1 所示。在这个平面上，若以 $-c$ 为基准，则 d 与凸包的切点在凸包上的斜率单调，故 $F'(k)$ 单调。* □

3.4.3 两者联系

令 $\mathcal{X}(S)$ 表示一个每一维坐标都是 0 或 1 的 $|E|$ 维空间内的点，其第 i 维坐标等于 1 当且仅当第 i 个元素属于 S。

令 $X(M_1, M_2) = \{\mathcal{X}(S) : S \in \mathcal{I}_1 \cap \mathcal{I}_2\}$。

令 $conv(X(M_1, M_2))$ 为包含点集 $X(M_1, M_2)$ 的最小的凸多胞形。

令

$$
\begin{aligned}
P(M_1, M_2) = \{x \in \mathbb{R}^{|E|} : \quad & x(S) \le \max_{T \in \mathcal{I}_1, T \subset S} |T| && \forall S \subset E \\
& x(S) \le \max_{T \in \mathcal{I}_2, T \subset S} |T| && \forall S \subset E \\
& x_e \ge 0 && \forall e \in E\}.
\end{aligned}
$$

定理 3.2. $conv(X(M_1, M_2)) = P(M_1, M_2)$。

该定理的证明详见 [2]。

定义 $|E|$ 维空间内的点 $\mathcal{X}(w)$ 满足 $\forall e \in E, \mathcal{X}(w)_e = w(e)$。那么拟阵交问题可以视为 $X(M_1, M_2)$ 中在 $-\mathcal{X}(w)$ 方向上的极点，同时也等于 $conv(X(M_1, M_2))$ 在 $-\mathcal{X}(w)$ 方向上的极点。

如果将 $P(M_1, M_2)$ 的限制视为线性规划的限制，权值函数 w 视为线性规划的目标函数。那么，我们可以将拟阵交问题转化为线性规划问题。

3.4.4 凸的性质

回归本题，令边集大小为 k 的最小内向森林问题的答案为 MDF(k)。我们有两个基于边集的拟阵，$\text{MDF}(k) = \min_{S \in \mathcal{I}_1 \cap \mathcal{I}_2, |S| \le k} w(S)$。

我们在 $P(M_1, M_2)$ 中加入关于 k 的限制得到 $P(M_1, M_2, k)$。

令

$$
\begin{aligned}
P(M_1, M_2, k) = \{x \in \mathbb{R}^{|E|} : \quad & x(S) \le \max_{T \in \mathcal{I}_1, T \subset S} |T| && \forall S \subset E \\
& x(S) \le \max_{T \in \mathcal{I}_2, T \subset S} |T| && \forall S \subset E \\
& x_e \ge 0 && \forall e \in E \\
& \textstyle\sum_{e \in E} x_e \le k && \}.
\end{aligned}
$$

引理 3.3. $P(M_1, M_2, k)$ 的极点全都是整点。

证明. 构造 $M_2^* = \left(E, \mathcal{I}_2^*\right)$，其中 $\mathcal{I}_2^* = \{S \in \mathcal{I}_2 : |S| \le k\}$。$M_2^*$ 是拟阵，所以 $conv(X(M_1, M_2^*)) = P(M_1, M_2^*)$。又因为 $P(M_1, M_2, k) = P(M_1, M_2^*)$，所以 $conv(X(M_1, M_2^*)) = P(M_1, M_2, k)$。 □

因此，加入大小不超过 k 的限制之后，依旧能忽略整数的限制，将问题转化为线性规划问题。令 $\mathrm{MDF}^*(i)$ 为以 $P(M_1, M_2, i)$ 为限制，以 w 为最优化函数的线性规划的值。那么，$\forall i \in \mathbb{Z}, \mathrm{MDF}^*(i) = \mathrm{MDF}(i)$。

将 $\sum_{e \in E} x_e \le k$ 置于线性规划之外，我们可得 $\mathrm{MDF}^{*\prime}(i)$ 是不减函数，所以 $\mathrm{MDF}^*(i)$ 是凸函数。因此，$\mathrm{MDF}(i)$ 也为凸函数，即 $\forall i \in \mathbb{Z}, \mathrm{MDF}(i+1) - \mathrm{MDF}(i) \le \mathrm{MDF}(i+2) - \mathrm{MDF}(i+1)$。

3.5 小结

本小节介绍了一些拟阵的知识，并将最小内向森林问题描述成拟阵交的形式。最后通过线性规划的知识得到了一个非常重要的结论：最小内向森林问题关于边数的函数是凸的。这个结论对后文有重要的帮助。

实际上，利用本节的内容，已经可以在多项式时间复杂度内解决最小内向森林问题。因为我们能将最小内向森林问题表示成拟阵交问题，并且可以在多项式时间复杂度内进行独立集的判断，所以可以直接套用带权拟阵交算法。但是复杂度过高，不推荐这种方法。

4 基于凸优化和最小树形图的算法

4.1 最小树形图问题

最小树形图问题可以被如下描述。

给定一张有向图 $G = (V, E)$，有边权 $w : E \mapsto \mathbb{R}$。对于给定的根 t，求权值最小的边子集，满足其边导出子图是一棵以 t 为根的内向树。

4.2 朱刘算法

“朱刘算法”的核心思想即为以下两个引理。

引理 4.1. 在最小树形图问题中，将一个结点的所有出边的权值同时加减，不影响最优解的树的形态。

证明. 任意一棵内向生成树的每个非根结点的出度都为 1，所以影响最优解的树的形态的因素只有每个结点出边的相对边权。 □

引理 4.2. 在最小树形图问题中，如果存在一个边权全为 0 的环，环上所有点的出边都非负，那么这个环可以视为一个结点。

证明. 这个引理非常抽象，具体来讲，该引理的意思是对于边权全为 0 的环，一定存在一种最优解使得恰有一条环上的边不被选。

因为内向树没有环，所以这个环上至少有一条边不被选。

如果存在一个最优解方案 H，使得这个环上有大于等于两条边不被选，令 S 为这些边的起点的点集，则在图 H 中，S 中至少有一个点到根的路径上不存在环上的点，令该点为 S_1。对于 H，将 $S - \{S_1\}$ 中每一个点的出边改成环上的边，得到 H^*，那么 H^* 一定是一棵内向生成树并且不劣于 H。如图 *3.1* 所示，左边是 H，右边是 H^*。 □

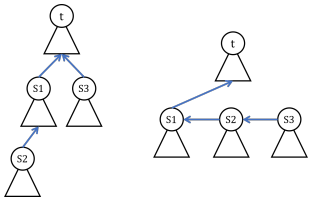


图 2: 图 3.1

利用以上两个引理，就可以得到朱刘算法，以下是对其的形式化的描述。

步骤(1) 任选一个待确定出边的结点 u。

步骤(2) 利用同时加减 u 的出边边权，将 u 的最小的出边边权变成 0，然后将这条边作为这个结点的出边。如果出边形成了环，则将这个环缩成一个结点。

步骤(3) 如果还有待确定出边的结点则返回 **步骤(1)**。

实现的时候，需要对每个结点建可并堆，支持全局加减并维护最小的出边，当多个结点缩成一个结点的时候将它们的堆合并，时间复杂度为 $O(E \log E)$。

4.3 最小内向森林问题的特殊情况

在 $k = |V| - 1$ 时，最小内向森林问题可以很方便地转化成最小树形图问题。

当 $k = |V| - 1$ 的时候，最小内向森林问题等价于要求一个不确定根的最小树形图问题。那么我们新建一个点 t 作为内向树的根，将 V 中的每一个点向其连一条权值一样且足够大的有向边。这样的话，任意一棵以 t 为根的最小内向树中，t 的入度都恰为 1，最后将这条边的权值减去即可。

4.4 凸优化

4.4.1 简介

凸优化是算法竞赛中常用的技巧，用来求解某个凸函数的某一项。

使用凸优化的条件是：对于任意的斜率，都能求出该斜率在凸函数上的切点。在算法竞赛中，求切点的方式往往是给凸函数增加一个一次函数后求全局的极值。

通过二分斜率，就能较为精确地求出凸函数上的某一点。

需要注意的是，凸优化只有在某些特定时候，才能求出凸函数上某一点的精确值。在凸优化只能求出近似值的时候，其精度可以达到指数级小。

4.4.2 在最小内向森林问题中的应用

由上述特殊情况受到启发，令 $T(\alpha)$ 表示，新建一个点 t 作为内向树的根，将 V 中的每一个点向其连一条权值为 α 的有向边后，最小树形图问题的答案。

考虑如何求解 $T(\alpha)$。让我们先枚举除了 t 结点的入边之外的边数，令其为 x，这一部分的最小代价为 $\mathrm{MDF}(x)$。此外，这 x 条边会形成 $|V|-x$ 棵彼此不连通的内向树，它们的根通过权值为 α 的边指向 t，这一部分的代价为 $(|V|-x)\alpha$。所以，$T(\alpha)$ 的值即为这两部分代价之和的最小值。

即：$T(\alpha)=\min\limits_{x\in[0,|V|)}\mathrm{MDF}(x)+(|V|-x)\alpha$。

由上一节的内容，我们可知，MDF 是一个凸函数，开口向上。我们观察这个式子，将括号展开，可以把 $T(\alpha)$ 看作是加上了斜率为 $-\alpha$ 后的最小值，也就是说可以求出 MDF 加上一个任意斜率的一次函数后的最小值。

加上一个斜率为 $-\alpha$ 的一次函数后的最小值，等同于凸函数在斜率为 α 上的切点。至此，我们就可以使用凸优化来求出 MDF 的某一项。

4.5 算法描述

我们得到了基于凸优化和最小树形图的算法，接下来对其形式化地描述。

步骤(1) 二分斜率为 α。

步骤(2) 新建结点 t，所有点向其连边权为 α 的有向边。

步骤(3) 从此时起，在 **步骤(4)** 与 **步骤(5)** 中，利用同时加减一个点的出边边权，使得每个点的最小出边边权始终为 0。

步骤(4) 任选一个待确定出边的结点 u。

步骤(5) 选择 u 的最小的出边作为这个结点的出边。如果出边形成了环，则将这个环缩成一个结点。

步骤(6) 如果还有待确定出边的结点则返回 **步骤(4)**。

步骤(7) 判断根节点的入度与 k 的关系，回到 **步骤(1)** 进行下一次二分。或达到目标精度退出。

4.6 小结

最小树形图算法的时间复杂度为 $O(E\log E)$，凸优化算法的时间复杂度为精度的对数。那么利用凸优化转化成最小树形图问题的时间复杂度为 $O(E\log E\log_{\frac{1}{2}}\varepsilon)$，其中 ε 为绝对精度误差除以边权的绝对值之和，即凸优化中的二分次数为 $\log_{\frac{1}{2}}\varepsilon$。

5 优先内向树扩张算法

5.1 避免新建结点

新建结点会失去很多直观的性质，所以考虑如何不新建结点。

令 3.5 中给出的算法的 **步骤(5)** 中，如果 u 选择出边后形成环并缩环，那么将这个操作称为内缩，否则称为扩张。令一组 **步骤(4)** 和 **步骤(5)** 称为一轮基础缩张操作。

观察到，任意结点 u 进行内缩操作之后，缩成的点依旧是一个待选择出边的结点。如果对于当前的任意一棵未选择过根的内向树，不停地选择这棵内向树的根，直到其扩张为止，那么可以得到从这棵内向树引出一条出边的最小代价，将其称为这棵内向树的扩张代价。具体来说，扩张代价为，从一棵内向树的根结点未进行过基础缩张操作，到内向树引出边的所有代价，包括最初根节点的最小出边变成 0 的代价。

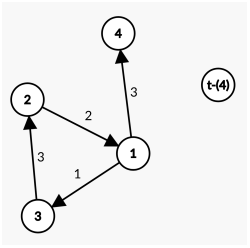
既然新建的结点 t 拥有每个点到其的权值相同且为 α 的边，但是新建结点的意义，并不仅限于当一棵内向树的扩张代价不小于 α 时，这棵内向树直接指向 t。

接下来给出一个例子来更好地说明这一点。

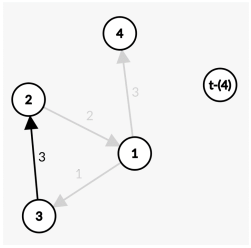
由图 4.1.1 所示，这是个四个点的图，现在新建了结点 t，且 $\alpha=4$，即所有点到 t 都有一条权值为 4 的有向边（图上未表示出来）。

先对 3 号点进行基础缩张操作，如图 4.1.2 所示。再对 2 号点进行基础缩张操作，如图 4.1.3 所示。此时，以 1 号点为根的内向树的扩张代价为 3，比 α 要小，但对其进行基础缩张操作后的结果却是指向 t 的。造成这种现象的原因在于，这棵内向树内，存在一个结点，它在内向树上的出边非常的大，导致存在先将其内缩至根再由该点指向 t 的可能。

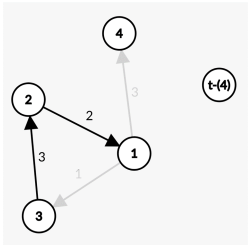
这使得我们需要引入一个概念：结点的预减值，记为 dec。



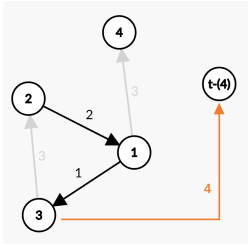
(a) 图 4.1.1



(b) 图 4.1.2



(c) 图 4.1.3



(d) 图 4.1.4

注意到，无论图上的边权如何变化，由同一个初始结点所引出的出边的变化值是一样的。被缩的原始结点并不是无端地消失了，而是对所有出边进行同时变化后，把出边和其他被缩点合并。那么，定义结点的预减值，为在当前状态下，将最小出边变成 0 后，每个原始结点的出边的减小值。

观察预减值在基础缩张操作中的实际表现，是将若干个集合合并，然后把合并完的集合内的所有预减值同时增加。对于缩完后的新点，我们定义这个新点的预减值为其所有结点的最大预减值。那么，在任意时候，一个结点 u 都有一条指向 t 的权值为 $\alpha - dec(u)$ 的边。

引理 5.1. 在最小内向森林问题中，给定一张带权有向图 $G = (V, E, w)$，在只进行基础缩张操作的情况下，形成了一棵内向树 H，且 H 的根的预减值是这棵内向树中最大的。那么，H 是其点导出子图的最小不定根内向生成树。

证明. 只需要证明，当新建结点 t，所有点向 t 连一条边权足够大且为 α 的边的时候，对 H 的根不停进行基础缩张操作，那么最终会选择根直接指向 t。假设根为 r，缩点后的根的点集为 R，那么内缩操作的实质只是：不停往 R 中加入预减值更小的元素并全局加。如此反复，R 中预减值最大的始终为 r。 □

引理 5.2. 在最小树形图问题中，如果根节点 t 有且仅有其他点指向它的边权为 α 的边。在只进行基础缩张操作的情况下，对于一棵内向树 H，它的根是树上 dec 最大的结点，那么对它进行基础缩张操作后会直接指向 t 当且仅当无视 t 后其扩张代价大于 α。

证明. 因为这样的 H 是其点导出子图中的最小不定根内向生成树。所以 H 要么对非 t 结点扩张，要么 H 的根直接指向 t。 □

那么，如果强制在根节点的预减值最大，那么 t 的意义就是对于扩张代价的限制。

得到 t 的作用后，如果我们能使得任意时刻每棵内向树的根都是预减值最大的点，那么就可以将 t 的限制显性地表示出来。要做到这一点，首先内缩操作不影响根节点的最大地位，所以只需要合理地改变扩张操作的顺序。对于两棵内向树 A, B，它们的根为 t_A, t_B，且 $dec(t_A) \le dec(t_B)$，将 A 指向 B，可以保持根的预减值最大。

对基础缩张操作稍加修改，增加对 dec 的优先级，即优先选择 dec 最小的结点。将这样的操作称为优先缩张操作。

引理 5.3. *在最小树形图问题中，如果根节点 t 有且仅有其他点指向它的边权为 α 的边。在只进行优先缩张操作的情况下，一棵内向树在最优解中会直接指向 t 当且仅当无视 t 后其扩张代价大于 α。*

证明. *只进行优先缩张操作的前提下，每一棵内向树的根的 dec 都是树中最大的。* □

现在可以形式化地描述不新建结点 t 的算法。

步骤(1) 二分斜率为 α。

步骤(2) 从此时起，在 **步骤(3)** 与 **步骤(4)** 中，利用同时加减一个点的出边边权，使得每个点的最小出边边权始终保持为 0，并维护预减值。

步骤(3) 选择预减值最小的待选择出边的结点 u，判断其扩张代价若大于 α，那么直接将答案增加 $\alpha - dec(u)$，去掉这棵内向树并跳过 **步骤(4)**。

步骤(4) 选择 u 的最小的出边作为这个结点的出边。如果出边形成了环，则将这个环缩成一个结点。

步骤(5) 如果还有待确定出边的结点则返回 **步骤(3)**。

5.2 避免凸优化

思考扩张代价与预减值的关系，发现，扩张代价即为扩张时的根的预减值。另一件容易发现的结论是，把内向树 A 接到内向树 B 上，内向树 B 的内缩操作不变。所以对于当前扩张代价最小的内向树，可以先进行该内向树的操作直至其扩张。于是，优先缩张的选择方式求得的内向森林等同于以下的选择方式。

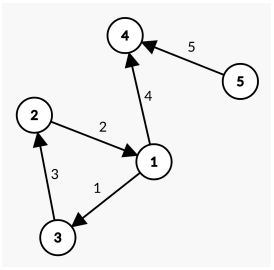
- 选择扩张代价最小的内向树，选择它的根 u。

也正因为把内向树 A 接到内向树 B 上，内向树 B 的内缩操作不变，所以内向树 B 的扩张代价不减。那么，在若干优先缩张操作中，所选的内向树的扩张代价只增不减。这时候，对于一个递增的操作序列，二分 α 就失去了意义。

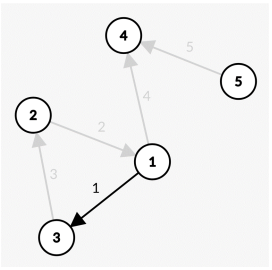
在去掉二分的同时，也不难发现，既然先前二分的位置是当前边集大小的最优解，那么这个过程中的任意时刻，都是当前边集的最优解。既然这样，那么不如顺便将任意的边集大小的答案同时求出。

5.3 具体算法与复杂度分析

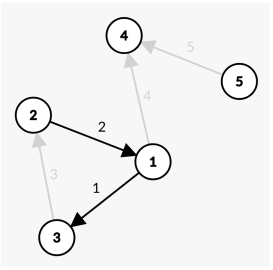
在形式化地描述之前，让我们看一个样例，回忆一下这个算法是如何运作的。



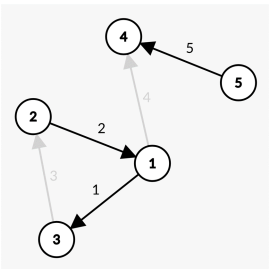
(e) 图 4.2.1



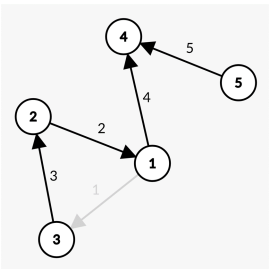
(f) 图 4.2.2



(g) 图 4.2.3



(h) 图 4.2.4



(i) 图 4.2.5

这个样例如图 4.2.1 所示，拥有 5 个点和 5 条边。起初每个点都是一棵独立的内向树，其中 1 号结点是扩张代价最小的结点，扩张后得到图 4.2.2。同理，2 号点扩张后得到图 4.2.3。

此时，拥有内向树 $\{1,2,3\}$、$\{4\}$ 和 $\{5\}$，他们的扩张代价分别为 6、$+\infty$ 和 5，故选择 5 号结点进行扩张，得到图 4.2.4。最后，选择内向树 $\{1,2,3\}$ 扩张，得到图 4.2.5。

5.3.1 算法描述

步骤(1) 起初有 $|V|$ 个内向树，每个内向树是一个点。

步骤(2) 令现在的边集为 E_{now}，记录 $\text{MDF}(|E_{\text{now}}|) = w(E_{\text{now}})$。

步骤(3) 求出现在每个内向树的扩张代价。

步骤(4) 如果没有内向树可以引出一条边则退出。

步骤(5) 选择扩张代价最小的内向树进行若干缩张操作直至扩张，返回 **步骤(2)**。

5.3.2 复杂度分析

暴力实现的瓶颈在于求每个内向树的扩张代价。因为把内向树 A 接到内向树 B 上，内向树 B 的内缩操作不变。所以可以提前处理每一棵内向树的所有内缩操作，这样对于一棵内向树，就能 $O(1)$ 的时间内求得其扩张代价。

可以使用可并堆来实现内缩操作并维护出边，这一部分时间复杂度为 $O(E\log E)$。为了方便地寻找扩张代价最小的内向树，还需要一个堆来维护内向树的扩张代价，这一部分时间复杂度为 $O(V\log V)$。

所以总时间复杂度为 $O(E\log E)$。

5.4 正确性的另一种证明

在第 2 节中，我们可以把最小内向森林问题描述成带权拟阵交问题。

定理 5.1. *对于任意一个带权拟阵交问题，该问题在大小为 k 时有解，那么对于任意一个大小为 $k-1$ 的最优解 B，都至少存在一个大小为 k 的最优解 A，使得：*

1. 如果对于所有 (u,v) 满足 $u\in B-A, v\in A-B$ 且 $B-\{u\}+\{v\}\in\mathcal{I}_1$ 但 $B+\{v\}\notin\mathcal{I}_1$，就将 u 与 v 之间连一条边。那么，$\exists x\in A-B$，满足 $B+\{x\}\in\mathcal{I}_1$，使得 $A-B-\{x\}$ 到 $B-A$ 有完美匹配。

2. 如果对于所有 (u,v) 满足 $u\in B-A, v\in A-B$ 且 $B-\{u\}+\{v\}\in\mathcal{I}_2$ 但 $B+\{v\}\notin\mathcal{I}_2$，就将 u 与 v 之间连一条边。那么，$\exists y\in A-B$，满足 $B+\{y\}\in\mathcal{I}_2$，使得 $A-B-\{y\}$ 到 $B-A$ 有完美匹配。

上述定理就是带权拟阵交算法的一部分，所以在此不作证明。

让我们回忆一下用来描述最小内向森林问题的两个拟阵，$\mathcal{I}_1$ 是视为无向边后无环的图，$\mathcal{I}_2$ 是每个点出度为 0 或 1 的图。

观察第 2 个拟阵。如果现在已知一个大小为 $k-1$ 的最优解 B，我们利用定理知道存在一个大小为 k 的最优解 A。对于一条 u 到 v 的边，由 $B+v\notin\mathcal{I}_2$ 可知，v 的起点是 B 中一条边的起点。同理，$A-B-y$ 中的所有边的起点都是 B 中某条边的起点。所以 A 相较于 B，在出边的数量上，恰有一个点从 0 变成了 1。

引理 5.4. *在最小内向森林问题中，如果已知最优解中，拥有出边的结点的集合为 S，那么对 S 中的每个元素依次执行基础缩张操作直至扩张即可得到最优解。*

证明. *既然 $V-S$ 中的每一个点都没有出边，那么将这个集合合并成一个结点 t，原问题就转化成了最小树形图问题。* □

那么，如果已知 B，想要求解 A。可以枚举是哪一个结点的出度从 0 变成了 1，求出对其执行基础缩张操作直至扩张的代价，最后取代价最小的结点即可。这样的代价等价于一棵内向树的扩张代价。因此，该过程与优先内向树扩张算法等价。

5.5 小结

至此，我们较为完美地解决了最小内向森林问题，进一步地，该算法可以对于所有的 k 同时求出答案。

并且，这个算法或许还有继续优化的空间，例如使用软堆等算法竞赛中不常用的数据结构。

6 应用

2020 Multi-University Training Contest 4 的 J 题是该算法的优秀应用之一。我作为此题的命题人，希望通过此题来普及最小内向森林问题的解决方法。

6.1 题目大意

给定 n 个谜题，标号从 1 到 n。要求将这些谜题分成 k 个非空序列。每个谜题都有一个权值 A，初始 $\forall i, A_i = 0$。

接下来进行 m 次判定，每次判定形如 (x, l, r, c)，表示如果 x 所在的序列中，存在 $y \in [l, r]$ 出现在 x 前，那么 $A_x = \max(A_x, c)$。

制定这 k 个非空序列的排列划分方案，最大化 $\sum_{i\in[1,n]} A_i$。

6.2 建图

建图，每个谜题建立一个点，在任意两点之间连接一条权值为 0 的边。对于每次判定，将 x 向所有 $[l, r]$ 中的点连一条边权为 c 的有向边。答案即为弱连通块数量为 k 的最大内向森林。

我们接下来证明这个建图的正确性，其分为两个部分。

第一个部分是：对于任意一个弱连通块数量为 k 的内向森林，都能找到对应的 k 个序列的划分方案。实际上只需要对于每个弱连通块，将任一逆拓扑序作为序列即可。

第二个部分是：对于任意一个 k 个序列的划分方案，都能找到对应的弱连通块数量为 k 的内向森林。那么只需要对于每个谜题 i，保留任一使 A_i 最大的一条边即可。

将边权取反后，我们就将原问题转化成了最小内向森林问题。

6.3 分析

观察到，在优先内向树扩张算法中，如果不遍历所有的自环，那么在遍历至多 $O(V)$ 条边后，该算法一定会停止。因为对于每一条非自环的边，如果内缩，那么点数减少；否则如果扩张，弱连通块数减少。所以我们的优化方向就是，在有限的时间内避免枚举过多的自环。

注意到在这张图上，虽然边数很多，如果使用 (x,l,r,c) 来描述一组边，图上的边的组数非常的少。于是对于每一个缩成的结点，用数据结构维护其初始点集，那么在询问这个点的出边时，可以非常快地找到 $[l,r]$ 中最小的不在点集里的点，以免枚举到自环。那么现在，除了每组边都会被至少被枚举一次以检验是否全是自环外，只会遍历 $O(V)$ 条边。

该算法的时间复杂度为 $O\left((n+m)\log(n+m)\right)$。

7 总结

本文围绕最小内向森林问题，先后介绍了拟阵、最小树形图问题和凸优化相关知识，并提出了优先内向树扩张算法，较好地解决了最小内向森林问题。并且，该算法代码简洁，实现细节较少，可以用来解决算法竞赛中的此类问题。在图论领域中，最小内向森林问题作为最小树形图问题的一个拓展，也有重要的理论价值。

参考文献

[1] 杨乾澜. 浅谈拟阵的一些拓展及其应用. IOI2018 中国国家候选队论文集, 2018.

[2] Michel X. Goemans. Massachusetts Institute of Technology 18.433, Combinatorial Optimization: Lecture notes on matroid intersection. MIT Mathematics, http://math.mit.edu/~goemans/18433S11/matroid-intersect-notes.pdf. pp. 7.

[3] Wikipedia contributors. Matroid. Wikipedia, https://en.wikipedia.org/wiki/Matroid.

[4] Wikipedia contributors. Edmonds' Algorithm. Wikipedia, https://en.wikipedia.org/wiki/Edmonds%27_algorithm.