arXiv:2302.05191v1 [cs.DM] 10 Feb 2023

识别置换德米坚科矩阵

Eranda Cela*
Vladimir Deineko†
Gerhard J. Woeginger‡

摘要

我们解决了德米坚科矩阵类的识别问题（RP）。我们的结果弥补了特殊结构矩阵识别领域中一个显著的空白。事实上，置换德米坚科矩阵的识别一直是一个悬而未决的问题；与此形成鲜明对比的是，德米坚科矩阵的重要子类，如 Kalmanson矩阵、Supnick 矩阵、蒙日矩阵和 Anti-Robinson 矩阵，其识别问题已得到高效解决。置换德米坚科矩阵的识别与难解的组合优化问题密切相关：当输入为德米坚科矩阵时，这些问题便可求解。德米坚科矩阵由 Demidenko 于 1976 年提出；当时他证明，若对称距离矩阵满足某些组合条件（如今称为*Demidenko* 条件），则旅行商问题（TSP）可在多项式时间内求解。在旅行商问题的背景下，识别问题是要判定是否存在一种城市重新编号方式，使重新编号后的距离矩阵满足 Demidenko 条件，从而使旅行商问题成为一个可在多项式时间内求解的特殊情形。我们证明，若存在这样的重新编号方式，就能在 $O(n^4)$ 时间内找到对 n 个城市的重新编号。

关键词。组合优化、旅行商问题、德米坚科条件、置换德米坚科矩阵。

1 引言

对排列进行优化是组合优化中的一个一般性问题。对于规模为 n 的实例，可行解集合 $\mathcal{F}_n$ 是 $\mathcal{S}_n$ 个关于 $\{1, 2, \ldots, n\}$、 $\mathcal{F}_n \subseteq \mathcal{S}_n$ 的排列所构成集合的子集。一般问题 P 可表示为 $\min\{f(\pi) : \pi \in \mathcal{F}_n\}$，其中$f$： $\mathcal{S}_n \to \mathbb{R}$ 是目标函数。组合优化中的一些基础 NP难问题可以用这种方式表述，例如旅行商问题（*TSP*）、二次指派问题（*QAP*）或路径旅行商问题（*PTSP*）。在所有这些情况下，目标函数 $f(\pi)$ 都由一个或两个矩阵确定

*cela@opt.math.tu-graz.ac.at。格拉茨工业大学离散数学系，Steyrergasse 30，A-8010 格拉茨，奥地利

†Vladimir.Deineko@wbs.ac.uk。华威商学院，华威大学，考文垂 CV4 7AL，英国

‡于2022年4月逝世。计算机科学一系（算法与复杂性），亚琛工业大学，扩建楼1号（2353），Ahornstr. 55，德国

系数。在 TSP 中，输入为一个 $n \times n$ 距离矩阵 $C = (c_{ij})$，目标函数 f_{TSP}^{C}定义为

$$f_{TSP}^{C}(\pi) := \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)} + c_{\pi(n)\pi(1)}\,, \text{ for } \pi \in \mathcal{F}_n := \mathcal{S}_n\,.$$

在二次指派问题中，输入由两个 $n \times n$ 矩阵 $A = (a_{ij})$、 $B = (b_{ij})$组成，目标函数 $f_{QAP}^{A,B}$ 定义如下

$$f_{QAP}^{A,B}(\pi) := \sum_{i=1}^{n} \sum_{i=1}^{n} a_{\pi(i)\pi(j)} b_{ij}\,, \text{ for } \pi \in \mathcal{F}_n := \mathcal{S}_n\,.$$

最后，在概率旅行商问题中，输入由一个 $n \times n$ 距离矩阵 $D = (d_{ij})$、一个起始索引i和一个结束索引 j i， $j \in \{1, 2 \ldots, n\}$组成，目标函数 f_{PTSP}^{D}定义如下

$$f_{PTSP}^{D}(\pi) := \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)}\,, \text{ for } \pi \in \mathcal{F}_n := \{\pi \in \mathcal{S}_n \colon \pi(1) = i\,, \pi(n) = j\}\,.$$

从理论和实践的角度来看，所有这些问题都很难。特别是，旅行商问题是组合优化和运筹学中研究最深入的问题之一，这不仅是因为它有众多实际应用，还因为它在上述领域开发和测试新方法方面具有特殊作用，参见例如 [1, 14, 17]。旅行商问题无法在多项式时间内精确求解（例如参见 [12]），且其近似求解为APX难（例如参见 [20]）。二次指派问题是一类经典且研究充分的问题，具有众多重要应用。该问题无法在多项式时间内精确求解，且近似求解为NP难，参见例如 [24], ；同时，从实践角度来看，该问题也非常困难。事实上，对于规模为 $n = 35$ 的一般实例，求得最优解仍然颇具挑战性，例如参见近期论文 [11]。最后，概率旅行商问题在理论上与前述问题同样困难，为NP难且APX难，例如参见 [26, 27]。

鉴于这些问题难以求解，对可解特殊情形进行刻画显然具有重要意义，并已成为一个成熟且活跃的研究领域。大多数可解特殊情形是通过对问题的系数矩阵施加某些组合条件而得到的，例如四点条件 （见 [7]）。这些条件会导出一些特殊的矩阵类，例如*Monge*、*Kalmanson*、*Supnick*、*Anti-Robinson* 和*Demidenko* 矩阵。旅行商问题中有相当多与这些矩阵类相关的可解特殊情形；全面综述请参见 [7] 及其中的参考文献。近年来，基于四点条件的特殊情形也已在二次指派问题和概率旅行商问题中得到研究，例如见 [4, 5]及其中的参考文献。

现在假设上述一般问题 P 在其系数矩阵（或多个系数矩阵）属于某个特定矩阵类（或多个矩阵类），例如 $\mathcal{C}$ （或 $\mathcal{C}$ 和 $\mathcal{D}$）时是可解的（可在多项式时间内求解）。

进一步地，考虑一个具有以下性质的 I P 实例： I 的系数矩阵（系数矩阵）可以根据某个排列 φ （或排列 φ、 ψ）进行置换，从而属于 $\mathcal{C}$。我们将具有置换后系数

矩阵（系数矩阵）的φ 实例称为置换后的实例。请注意，如果已知排列 φ （排列 φ、 ψ），或者可以高效地计算出该排列，则置换后的实例是易处理的。在这种情况下，利用排列 φ （φ 和 ψ），即可根据置换后实例的最优解轻松获得原始实例 I 的最优解。下面以旅行商问题为例说明这一思想。假设旅行商问题中距离矩阵属于 $\mathcal{C}$ 的特殊情形是多项式时间可解的。考虑一个距离矩阵为 C 的旅行商问题实例，该矩阵可以通过某个排列 φ进行置换，使得 $C^{\varphi} := (c_{\varphi(i)\varphi(j)}) \in \mathcal{C}$。设 π^* 为距离矩阵为 C^{φ}的旅行商问题实例的一个最优解。则 $\varphi \circ \pi^*$ 是距离矩阵为 C的旅行商问题实例的一个最优解，因为对于任意 $\pi \in \mathcal{S}_n$，以下等式成立：

$$f_{TSP}^{C^{\varphi}}(\pi) = \sum_{i=1}^{n-1} c^{\varphi}_{\pi(i)\pi(i+1)} + c^{\varphi}_{\pi(n)\pi(1)} = \sum_{i=1}^{n-1} c_{\varphi(\pi(i))\varphi(\pi(i+1))} + c_{\varphi(\pi(n))\varphi(\pi(1))} = f_{TSP}^{C}(\varphi \circ \pi)$$

因此，是否存在一个排列 φ 使得 $C^{\varphi} \in \mathcal{C}$ 成立这一问题，与所考虑的旅行商问题实例能否高效求解有关。若能回答这一问题，并在答案为肯定时确定相应的排列，就可以扩大该问题可在多项式时间内求解的特殊情形范围。

一类 $\mathcal{C}$ 矩阵（可通过某些组合性质来描述）的通用识别问题定义如下：

问题： RECOGNITION-$\mathcal{C}$
实例： $n \in \mathbb{N}$，一个 $n \times n$ 矩阵 C。
任务： 是否存在某个排列 $\varphi \in \mathcal{S}_n$，使得排列后的矩阵 C^{φ} 属于 $\mathcal{C}$? 若存在，则确定这样一个 φ。

特别是，Demidenko 矩阵类的识别问题与以下两个结果相关。1976 年，Vitali Demidenko [10] 证明，限于 Demidenko 矩阵的旅行商问题可在 $O(n^2)$ 时间内求解（参见 Gilmore、Lawler 和 Shmoys [13]）。近期论文 [5] 表明，Demidenko 矩阵上的概率旅行商问题也可在多项式时间内求解。

尽管 Demidenko 矩阵这一类的识别问题悬而未决已约四十年，但对于该类中的若干真子类，已知其识别问题可在多项式时间内求解，例如 Kalmanson矩阵、Supnick 矩阵、蒙日矩阵和 Anti-Robinson 矩阵，见 [6, 8, 9, 19, 21]。所有这些矩阵类都使排列上的组合优化问题出现了可在多项式时间内求解的情形，其中包括旅行商问题、二次指派问题和/或概率旅行商问题，例如见 [2, 3, 4, 5,7, 9, 13, 15, 16, 18, 22, 25]。本文填补了这一空白，解决了 Demidenko 矩阵的识别问题。更确切地说，我们展示了如何判定给定的 $n \times n$矩阵 D 能否通过排列变为 Demidenko 矩阵，并在答案为肯定时，如何在 $O(n^4)$ 时间内构造相应的排列。

论文结构。下一节将介绍一些基本记号，并定义 Demidenko 矩阵和 Anti-Robinson 矩阵；事实证明，这些矩阵

与本文相关。在第 3 节中，我们将讨论（经置换的）Demidenko 矩阵和（经置换的）Robinson 矩阵的一些性质，并重点探讨这些矩阵类之间的关系。接着，在第 4 节中，我们将介绍用于 Recognition-$\mathcal{D}$ 的算法，即针对 $\mathcal{D}$ Demidenko 矩阵类的识别问题。最后，我们将在第 5 节中简要总结全文。

2 定义与记号

定义 2.1 若对称 $n \times n$ 矩阵 $C = (c_{ij})$ 的元素满足以下不等式，则称其为 Demidenko 矩阵

$$c_{ji} + c_{kl} \le c_{jl} + c_{ki}\,, \qquad for\ all\ 1 \le i < j < k < l \le n. \tag{1}$$

若存在一个对其行和列进行排列的排列 φ，使得排列后的矩阵 $C^{\varphi} = (c_{\varphi(i)\varphi(j)})$ 是 *Demidenko* 矩阵，则称矩阵 $C = (c_{ij})$ 为 经过排列的 *Demidenko* 矩阵。这样的排列 φ 称为矩阵 C的 Demidenko 排列 。

请注意，对于 $i \in \{1, 2, 3, \ldots, n\}$，条件 (1) 分别不涉及元素 c_{ii}、c_{1i} 和 c_{ni}。

系统 (1) 包含 $O(n^4)$ 个不等式，但很容易看出，它等价于以下由 $O(n^3)$ 个不等式组成的系统

$$c_{ji} + c_{j+1,l} \le c_{jl} + c_{j+1,i}\,, \qquad \text{for all } 1 \le i < j < j+1 < l \le n. \tag{2}$$

此外，很容易看出，后一个系统 (2) 等价于以下条件，可在 $O(n^2)$ 时间内完成检验

$$\max_{i=1,\ldots,j-1} \{c_{ji} - c_{j+1,i}\} \le \min_{l=j+2,\ldots,n} \{c_{jl} - c_{j+1,l}\}\,, \qquad \text{for all } \ 2 \le j \le n-2. \tag{3}$$

因此，可以在 $O(n^2)$ 时间内判定给定的对称 $n \times n$ 矩阵 C 是否为 Demidenko 矩阵。

接下来我们定义 Robinson 矩阵。它们源自组合数据分析 [23]，并且在组合优化问题的背景下有一些很好的应用和意义，参见例如 [4, 18, 21]。

定义 2.2 如果对称 $n \times n$ 矩阵 $A = (a_{ij})$ 满足以下条件，则称其为反罗宾逊矩阵：

$$a_{ik} \ge \max\{a_{ij}, a_{jk}\}\,, \qquad for\ all\ 1 \le i < j < k \le n. \tag{4}$$

换言之，在 A 的每一行和每一列中，沿着远离主对角线的方向移动时，元素不会递减。若存在一个对其行和列进行排列的排列 φ，使得排列后的矩阵 $C^{\varphi} = (c_{\varphi(i)\varphi(j)})$ 是反罗宾逊矩阵，则称矩阵 $C = (c_{ij})$ 为一个排列后的反罗宾逊矩阵。这样的排列 φ 称为该矩阵的反罗宾逊排列 C。

请注意，反罗宾逊矩阵构成 Demidenko 矩阵类的一个子类。事实上，不等式 (4) 蕴含 $a_{ji} \le a_{ki}$ 和 $a_{kl} \le a_{jl}$，其中 $1 \le i < j < k < l \le n$；将这些不等式相加即可得到不等式 (1)。很容易验证上述包含关系是严格的，也就是说，存在不是反罗宾逊矩阵的 Demidenko 矩阵。例如，参见下面的 5×5 Demidenko 矩阵，并注意，对于任意 $n \in \mathbb{N}$, 都可以构造出类似的 $n \times n$ 矩阵。

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

由于 $a_{13} < a_{12}$, A 不是反罗宾逊矩阵。

记号。下面将介绍本文中会用到的一些记号。对于给定的 $n \times n$ 矩阵 C，我们用相应的索引标记其每一行（列）。因此，$I = \{1, \dots, n\}$ 是 C的行（列）集合。若行 i 先于 行 j 出现在C（简记为 $i \prec j$），则称行 i 先于行 j，其中行 i 在 C中出现在行 j 之前。若对于所有 $k_1 \in K_1$ 和所有 $k_2 \in K_2$，都有 $k_1 \prec k_2$，则称行（列）集合 K_1 先于行（列）集合 K_2。此时记作 $K_1 \prec K_2$。设 $V = \{v_1, v_2, \dots, v_r\}$ 是 I的子集。我们用 $C[V]$表示 C 的 $r \times r$ 子矩阵，该子矩阵由删除不属于 V的所有行和列得到。

对于一个排列 $\pi \in \mathcal{S}_n$，若对于所有$i \in \{1, 2, \dots, n\}$，$\pi(i) = x_i$ 成立，则记作 $\pi = \langle x_1, x_2, \dots, x_n \rangle$。对于两个指标子集 K、$L \subset I$，其中 $K \cap L = \emptyset$，以及一个排列π，若对于任意 $k \in K$ 和任意 $l \in L$，指标 i、$j \in I$ 满足 $\pi(i) = k$、$\pi(j) = l$，且 $i < j$，则称 K 在 π 中先于 L。此时记作 $K \prec_\pi L$，或简记为 $K \prec L$（当排列 π可由上下文明确时）。最后，设 $I = \{1, 2, \dots, n\}$ 和$K \subset I$满足 $K = \{s_1, s_2, \dots, s_{|K|}\}$，并且 $1 < |K| < n$和$s_1 < s_2 < \dots < s_{|K|}$。若对于任意$i$，$j \in \{1, 2, \dots, |K|\}$、$\tau(s_i) < \tau(s_j)$ 蕴含 $\sigma(i) < \sigma(j)$，反之亦然，则称排列 $\tau \in \mathcal{S}_n$ 和排列 $\sigma \in \mathcal{S}_{|K|}$ 在 K 上重合。

3 置换德米坚科矩阵与置换 Anti-Robinson 矩阵之间的关系

本节探讨（置换）Demidenko 矩阵与（置换）Anti-Robinson 矩阵之间的关系。下一个引理描述了某些特殊 Demidenko 矩阵与 Anti-Robinson 矩阵之间的关系。

引理 3.1 设 C 是一个 $n \times n$ *Demidenko* 矩阵，使得 $c_{1j} = c_{i1} = a$ 对于某个 $a \in \mathbb{R}$和任意 i, $j \in I := \{2 \dots, n\}$成立。进一步，假设对于某个 $r \in \{1, 2, \dots, n-2\}$，存在常数 $b \in \mathbb{R}$ 和指标子集 $K \subseteq I' := I \setminus \{1, 2, \dots, r, n\}$，使得$\sum_{j=1}^{r} c_{ij} - r c_{in} = b$ 对任意 $i \in K$成立。那么子矩阵 $C[K]$ 是一个反罗宾逊矩阵。证明。首先考虑任意两个指标 $i_1, i_2 \in K$，$i_1 < i_2$ 并注意到

$$\sum_{j=1}^{r} c_{i_1 j} - r c_{i_1 n} = \sum_{j=1}^{r} c_{i_2 j} - r c_{i_2 n} \tag{5}$$

由此可得 $c_{i_1 j} - c_{i_1 n} = c_{i_2 j} - c_{i_2 n}$ 对任意 $j \in \{1, 2, \ldots, r\}$成立。事实上，假设这些等式中至少有一个不成立，并将 j_1 记为 $\{1, 2, \ldots, r\}$ 中满足$c_{i_1 j_1} - c_{i_1 n} \neq c_{i_2 j_1} - c_{i_2 n}$的最小指标。四元组 $j_1 < i_1 < i_2 < n$的 Demidenko 条件推出 $c_{i_1 j_1} - c_{i_1 n} < c_{i_2 j_1} - c_{i_2 n}$。后一个不等式与 5 结合，意味着存在一个 $j_2 \in \{1, 2, \ldots, r\}$，使得 $c_{i_1 j_2} - c_{i_1 n} > c_{i_2 j_2} - c_{i_2 n}$ 成立。但这违反了四元组 $j_2 < i_1 < i_2 < n$的 Demidenko 条件。特别地，对于 $j = 1$，记 $c_{i_1 1} - c_{i_1 n} = c_{i_2 1} - c_{i_2 n} =: b'$ 对任意 $i_1, i_2 \in K$成立。后面的这些等式与$c_{i_1 1} = c_{i_2 1} = a$ 一起推出 $c_{in} = c_{ni} = a - b'$ 对任意 $i \in K$成立。现在考虑三元组 $j, k, l \in K$，其中 $j < k < l$，并将 $i = 1$代入 (1)：$c_{j1} + c_{kl} \leq c_{k1} + c_{jl}$。取 $c_{j1} = c_{k1}$，得到$c_{kl} \leq c_{jl}$。类似地，对于满足 $i < j < k$的任意三元组 $i, j, k \in K$，将 $l = n$代入 (1)：$c_{ji} + c_{kn} \leq c_{jn} + c_{ki}$。取 $c_{kn} = c_{jn} = a - b'$，得到 $c_{ji} \leq c_{ki}$。因此，子矩阵 $C[K]$满足定义 (2.2) 中的不等式 (4)，故它是一个反罗宾逊矩阵。□

上述引理所述的关系可推广为置换 Demidenko 矩阵与置换 Anti-Robinson 矩阵之间的关系，具体如下。

引理 3.2 设 C 是一个 $n \times n$ 置换 ***Demidenko*** 矩阵，且 $\tau \in \mathcal{S}_n$ 是 C的 ***Demidenko*** 排列。假设 τ满足 $\tau(n) = q$和 $\tau(j) = p_j$，其中 $j \in \{1, 2, \ldots, r\}$及某个 $r \leq n - 2$满足条件，且 p_1、p_2、……、p_r、$q \in I := \{1, 2, \ldots, n\}$是两两不同的指标。此外，设 C 的首行（列）为常数，即对于某个 $a \in \mathbb{R}$及任意 $i \in \{2, \ldots, n\}$，$c_{1i} = c_{i1} = a$ 成立。进一步假设存在一个常数 $b \in \mathbb{R}$和一个指标子集 $K \subseteq I' := I \setminus \{p_1, \ldots, p_r, q\}$，使得对于任意 $i \in K$，$\sum_{j=1}^{r} c_{\tau(i) p_j} - r c_{\tau(i) q} = b$成立。那么，子矩阵 $C[K]$ 是一个置换 ***Anti-Robinson*** 矩阵。特别地，在 K 上与 τ 一致的唯一排列 $\sigma \in \mathcal{S}_{|K|}$ 是$C[K]$的 ***Anti-Robinson*** 排列。

证明。考虑 Demidenko 矩阵 $C^\tau = (c_{ij}^\tau)$，其中 $c_{ij}^\tau = c_{\tau(i)\tau(j)}$ 对任意 i，$j \in I$均成立。显然，其首行和首列中的所有非对角线元素都等于 a。此外，对于任意$i \in K$，我们有 $\sum_{j=1}^{r} c_{ij}^\tau - r c_{in}^\tau = \sum_{j=1}^{r} c_{\tau(i) p_j} - r c_{\tau(i) q} = b$。引理 3.1 表明，矩阵 $C^\tau[K^\tau]$是一个反罗宾逊矩阵。现在注意到，$C^\tau[K^\tau]$是由 $C[K]$通过按照唯一的排列 $\sigma \in \mathcal{S}_{|K|}$ 对其行和列进行排列得到的，该排列在 K上与 τ 一致。□

接下来注意，我们可以将给定的 Demidenko 矩阵直接变换为首行（以及首列）为常数的 Demidenko 矩阵。

观察3.3 设 C 为一个 $n \times n$ ***Demidenko*** 矩阵。则矩阵 $C' = (c'_{ij})$对于 c'_{ij}：
$= c_{ij} - c_{1j} - c_{i1}$任意 i、$j \in I := \{1, 2, \ldots, n\}$，都是 ***Demidenko*** 矩阵，其中$c'_{i,\ 1} = c'_{1,\ i} = -c_{11}$ 对于任意 $i \in I$。

设 C 为给定的对称 $n \times n$ 矩阵。目标是确定是否存在一个适用于矩阵 C的 Demidenko 排列 τ，即一个排列 τ，使得 $C^\tau = (c_{\tau(i)\tau(j)})$ 是 Demidenko 矩阵；如果存在，还要计算 τ。为此，我们将确定适用于矩阵 C的 Demidenko 排列的一些简单组合性质。

引理 3.4 设 C为一个对称 $n \times n$矩阵，并设 p_1、p_2、…、p_r、$q \in I := \{1, 2, \ldots, n\}$是 $r \leq n - 2$的两两不同指标。设 $I' = I \setminus \{p_1, \ldots, p_r, q\}$ 和 $m :=$ 最小值 $\{\sum_{j=1}^{r} c_{ip_j} - r c_{iq} : \ i \in I'\}$。进一步设 $K := \{i \in I' : \ \sum_{j=1}^{k} c_{ip_j} - k c_{iq} = m\}$。对于 C 的任意 ***Demidenko*** 排列 τ，使得 $\tau(j) = p_j$，对于 $1 \leq j \leq r$，且 $\tau(n) = q$（如果存在这样的排列），以下陈述成立：

(i) 如果对于某个 $s \in I'$ 有 $K = \{s\}$，则 $\tau(r+1) = s$。

(ii) 如果 $1 < |K| < n - r - 1$，那么 $K \prec L$ 属于 τ，其中 $L := I' \setminus K$。换言之，$\{\tau(i) \colon\ i \in \{r+1, \ldots, r+|K|\}\} = K$ 成立。

证明。由于 C^τ 是一个 Demidenko 矩阵，且对于 $1 \le j \le r$，$\tau(j) = p_j$，以及 $\tau(n) = q$，我们将 (1) 应用于 $i \in \{1, 2 \ldots, r\}$、j，$k \in I'$，使得 $j < k$、$l = n$，从而得到

$$c^\tau_{ji} - c^\tau_{jn} \le c^\tau_{ki} - c^\tau_{kn} \text{ or equivalently } c_{\tau(j)p_x} - c_{\tau(j)q} \le c_{\tau(k)p_x} - c_{\tau(k)q}\,.$$

这些不等式意味着

$$\sum_{x=1}^{r} c_{\tau(j),p_x} - r c_{\tau(j),q} \le \sum_{x=1}^{r} c_{\tau(k),p_x} - r c_{\tau(k),q} \text{ for any } k > j,\, k, j \in I'\,. \tag{6}$$

不等式 (6) 意味着 $\tau(r+1) \in K$。于是，对于 $K = \{s\}$，我们得到 $\tau(r+1) = s$，因此 (i) 成立。

现在假设 $1 < |K| \le n - 2$。不等式 (6) 表明，对于任意 $s \in K$ 和任意 $s' \in L = I' \setminus K$，$\{s\}$ 在 τ 中位于 $\{s'\}$ 之前。因此，$K \preceq L$ 属于 τ，且 (ii) 成立。□

引理 *3.5* 设 C 是一个对称的 $n \times n$ 矩阵，且 $p_1, p_2, \ldots, p_r, q \in I := \{1, 2, \ldots, n\}$ 是满足 $1 \le r \le n-3$ 的某个 r 的两两不同的指标。设 $m := \min\{\sum_{x=1}^{r} c_{ip_x} - r c_{iq} \colon\ i \in I' := I \setminus \{p_1, \ldots, p_r, q\}\}$，$K := \{i \in I' \colon\ \sum_{x=1}^{r} c_{ip_x} - r c_{iq} = m\}$ 和 $L := I' \setminus K$。设 $C' = (c'_{ij})$ 按 $c'_{ij} = c_{ij} - c_{ip_1} - c_{p_1 j}$ 定义，其中 $i, j \in I$。进一步假设存在一个关于 C 的 *Demidenko* 排列 τ，使得对于 $j \in \{1, 2, \ldots, r\}$，$\tau(j) = p_j$，且 $\tau(n) = q$。

(i) 如果 $1 < |K| = n - (r+1)$，那么 $C'[K]$ 是一个置换 *Anti-Robinson* 矩阵。此外，对于相对于 $C'[K]$的任意 *Anti-Robinson* 排列 π，与 π 在 K 上重合且满足 $\tau'(x) = p_x$（对于 $x \in \{1, 2, \ldots, r\}$）以及 $\tau'(n) = q$的排列 $\tau' \in \mathcal{S}_n$，是 C的一个 *Demidenko* 排列。

(ii) 假设 $1 < |K| < n-(r+1)$。考虑对称的 $(|K|+1)\times(|K|+1)$ 矩阵 $D = (d_{ij})$，它由 $C'[K]$ 添加一个任意选取的 $(|K|+1)$-*st* 列（行）得到，其中 $d_{|K|+1,\,|K|+1}$ 和 $d_{i,\,|K|+1} = d_{|K|+1,\,i} = M \sum_{j \in L} c_{ij}$ 分别对应 $i \in \{1, 2, \ldots, |K|\}$ 和M，且后者是一个足够大的正数。于是，D 是一个置换 *Anti-Robinson* 矩阵，而矩阵 D 的一个 *Anti-Robinson* 排列 σ' 可通过对任意 $i \in \{1, 2, \ldots, |K|\}$取 $\sigma'(|K|+1) = |K|+1$和 $\sigma'(i) = \sigma(i)$得到，其中 σ 是唯一一个与 τ 在 K上重合的排列。

反之，对于矩阵 D 的任意 *Anti-Robinson* 排列 π'，都存在矩阵 C 的 *Demidenko* 排列 τ'，使得 $\tau'(x) = p_x$对于 $x \in \{1, 2, \ldots, r\}$, $\tau'(n) = q$成立，并且 τ' 在 K 上与 π' 重合。

(i) 的证明。首先假设 $|K| = n - (r+1)$。那么，根据引理 3.2，$C'[K]$ 是一个置换 Anti-Robinson 矩阵，而唯一确定的排列 $\sigma \in \mathcal{S}_{|K|}$ 在 K 上与 τ 重合，且是矩阵 $C'[K]$ 的 Anti-Robinson 排列。

现在考虑矩阵 $C'[K]$ 的任意 Anti-Robinson 排列 π，并令 $\tau' \in \mathcal{S}_n$ 为唯一确定的排列：它在 K 上与 π 一致，且满足 $\tau'(i) = p_i$，其中

$1 \le i \le r$，且 $\tau'(n) = q$。不难验证，经 τ' 排列后的 C' 满足 Demidenko 条件 (2)。事实上，对于 I' 中任意的 i、j、l，若 $r < i < j < j+1 < l < n$，则有

$$c'_{\tau'(j)\tau'(i)} \le c'_{\tau'(j+1)\tau'(i)} \text{ and } c'_{\tau'(j+1)\tau'(l)} \le c'_{\tau'(j)\tau'(i)}\,, \tag{7}$$

因为 τ' 在 K 上与 π 重合，且经 π 置换的 $C'[K]$ 是一个反罗宾逊矩阵。在此情况下，不等式 (7) 显然蕴含 (2)。此外，对于任意 $i \le r$，以及任意 j，$l \in I'$，当 $i < j < j+1 < l < n$ 时，(7) 中左侧的不等式取等号；而对于任意 i，$j \in I'$，当 $r < i < j < j+1 < l = n$ 时，(7) 中右侧的不等式取等号。最后，对于任意 $i \le r$ 以及任意 $j \in I'$，当 $i < j < j+1 < l = n$ 时，(7) 中的两个不等式均取等号。因此，对于任意 i，j，l，在 I 中，当 $1 \le i < j < j+1 < l \le n$ 时，经 τ' 置换的 C' 满足 Demidenko 条件，从而完成了 (i) 的证明。

(*ii*) 的证明。现在考虑 $1 < |K| < n-(r+1)$ 的情形。令 $K = \{s_1, s_2, \ldots, s_{|K|}\}$ 和 $s_1 < s_2 < \ldots < s_{|K|}$。考虑矩阵 C 的 Demidenko 排列 τ，其中对于 $x \in \{1, 2, \ldots, r\}$ 有 $\tau(x) = p_x$，对于 τ 有 $n(= q)$。那么显然，τ 也是 C' 的 Demidenko 排列。此外，对于任意 $i \in I$，都有 $c'_{\tau(1)i} = c'_{i\tau(1)} = -c_{pp}$。进一步地，由 K 的定义，对于任意 $i \in K$，都有 $\sum_{x=1}^{r} c'_{\tau i p_x} - r c'_{\tau(i)q} = m$。于是，引理 3.2 表明，$C'[K]$ 是一个置换 Anti-Robinson 矩阵，而唯一确定的 $\sigma \in \mathcal{S}_{|K|}$（其在 K 上与 τ 一致）是 $C'[K]$ 的一个 Anti-Robinson 排列。现在考虑矩阵 D，即 $(|K|+1) \times (|K|+1)$ 矩阵。我们证明它是一个置换 Anti-Robinson 矩阵。更确切地说，我们证明，由下式给出的排列 $\sigma' \in \mathcal{S}_{|K|+1}$：对于任意 $i \in \{1, 2, \ldots, |K|\}$，$\sigma'(i) = \sigma(i)$，以及 $\sigma'(|K|+1) = |K|+1$，是 D 的一个 Anti-Robinson 排列。为此，只需证明以下两族不等式

$$d_{\sigma'(i+1),|K|+1} \le d_{\sigma'(i),|K|+1} \text{ for any } i \in \{1, 2, \ldots, |K|-1\} \tag{8}$$

and

$$d_{\sigma'(i)\sigma'(|K|)} \le d_{\sigma'(i),|K|+1} \text{ for any } i \in \{1, 2, \ldots, |K|-1\}\,. \tag{9}$$

注意，对于任意 $i \in \{1, 2, \ldots, |K|-1\}$，我们有

$$d_{\sigma'(i+1),|K|+1} = d_{\sigma(i+1),|K|+1} = d_{\tau(s_{i+1}),|K|+1} = M \sum_{j \in L} c'_{\tau(s_{i+1})\tau(j)}\,, \tag{10}$$

类似地

$$d_{\sigma'(i),|K|+1} = M \sum_{j \in L} c'_{\tau(s_i)\tau(j)}\,. \tag{11}$$

由于 $K \preceq L$，且矩阵 $(C')^{\tau}$（通过对 C' 进行 τ 排列得到）是首列为常数的 Demidenko 矩阵，因此，带有 $1 < i < i+1 < j$ 的 $(C')^{\tau}$ 的 Demidenko 条件蕴含 $c'_{\tau(s_{i+1})\tau(j)} \le c'_{\tau(s_i)\tau(j)}$ 对任意 $j \in L$ 均成立。后一个不等式与 (10) 和 (11) 一起蕴含 (8)。最后，(9) 等价于 $c'_{\tau(s_i)\tau(s_{|K|})} \le M \sum_{j \in L} c'_{\tau(s_i)\tau(j)}$，而只要将 M 选得足够大，就能保证后一个不等式成立。

现在假设 D 是一个反罗宾逊矩阵，并考虑矩阵 D 的一个 Anti-Robinson 排列 $\pi' \in \mathcal{S}_{|K|+1}$。将 M 选得足够大，即可确保 $\pi'(|K|+1) = |K|+1$。于是，对于任意 $i \in \{1,2,\ldots,|K|\}$，带有 $\pi(i) = \pi'(i)$ 的 $\pi \in \mathcal{S}_{|K|}$ 是 $C'[K]$ 的一个 Anti-Robinson 排列。考虑满足 $\tau(j) = p_j$（对 $j \in \{1,2,\ldots,r\}$）且 $\tau(n) = q$ 的 C 的 Demidenko 排列 τ；根据引理的假设，该排列存在。显然，τ 也是矩阵 C' 的一个 Demidenko 排列。因此，引理 3.4 蕴含 $K \preceq L$ 属于 τ，即 $\{\tau(i): i \in K\} = \{r+1,\ldots,,|K|+r\}$ 成立。若 τ 在 K 上与 π 一致，则结论成立。否则，我们通过重新排列条目 $\tau(s_i)$、$i \in K$ 来修改 τ，从而得到一个新排列 $\tau' \in \mathcal{S}_n$，它在 K 上与 π 一致，并满足 $\tau'(j) = p_j$（对 $j \in \{1,2,\ldots,r\}$），且 $\tau'(n) = q$。只需证明 τ' 是 C' 的一个 Demidenko 排列，因而也是 C 的一个 Demidenko 排列，即可完成证明。

不失一般性，假设 τ' 是通过对 τ 进行换位 $(s_i,\ s_j)$ 得到的，其中$i,\ j \in \{1,2,\ldots,|K|\}$，且 $i \neq j$。这意味着，对于任意 $l \in I \setminus \{s_i, s_j\}$，$\tau'(s_i) = \tau(s_j)$、$\tau'(s_j) = \tau(s_i)$，以及$\tau(l) = \tau'(l)$。我们证明，$C'$的行（列）$s_i$和 s_j除对角线元素外完全相同；因此，这种相同意味着 τ'是 C' 的 Demidenko 排列，也同样是 C的 Demidenko 排列。

反设存在一个 $l \in I \setminus \{s_i, s_j\}$，使得 $c'_{s_i,l} \neq c'_{s_j,l}$。令$\sigma' \in \mathcal{S}_{|K|+1}$ 和 $\sigma \in \mathcal{S}_{|K|}$ 按照 (ii) 已证明方向中的方式由 τ 定义。因此，τ 在 K上与 σ 一致，σ 是矩阵 $C'[K]$的 Anti-Robinson 排列，且σ' 是矩阵 D的 Anti-Robinson 排列。由于 τ' 是通过对 τ 进行换位 (s_i,s_j) 得到的，因此，σ 由 π 得到，而 σ' 由 π' 得到，二者分别通过换位 (i,j) 实现。注意到 $l \notin K$。事实上，在假设 $l \in K$成立且不失一般性地假设 $\tau(s_i) < \tau(s_j) < \tau(l)$ 的条件下，可得 $c'_{\tau(s_j)\tau(l)} \leq c'_{\tau(s_i)\tau(l)}$ 且 $c'_{\tau(s_i)\tau(l)} = c'_{\tau'(s_j)\tau(l)} \leq c'_{\tau'(s_i)\tau(l)} = c_{\tau(s_i)\tau(l)}$，因此 $c'_{\tau(s_i)\tau(l)} = c'_{\tau(s_j)\tau(l)}$，这与 $c'_{s_i,l} \neq c'_{s_j,l}$矛盾。因此 $l \notin K$，从而推出 $l \in L$。由于 π'和 σ' 均为 D 的 Anti-Robinson 排列，且对任意 $k \notin \{i,j\}$，$\pi'(k) = \sigma'(k)$，由类似论证可得 $d_{i,|K|+1} = d_{j,|K|+1}$，或等价地，$\sum_{t\in L} c'_{\tau(s_i)\tau(t)} = \sum_{t\in L} c'_{\tau(s_j)\tau(t)}$。不失一般性，假设 $c'_{\tau(s_i)\,\tau(l)} < c'_{\tau(s_j)\,\tau(l)}$。于是，存在一个 $l' \in L \setminus \{l\}$，使得 $c'_{\tau(s_i)\,\tau(l')} > c'_{\tau(s_j)\,\tau(l')}$。由于 $l \neq l'$，因此$\tau(l) = \tau'(l) \neq \tau'(l') = \tau(l')$。不失一般性，假设 $\tau(l) < \tau(l')$。那么，对于 $(C')^\tau$，指标 $\tau(s_i) < \tau(s_j) < \tau(l) < \tau(l')$ 违反了 Demidenko 条件 (1)，这与将 τ 选为相对于 C'的 Demidenko 排列相矛盾。类似地，假设$\tau(l') < \tau(l)$ 会与 τ'是 C'的 Demidenko 排列相矛盾。因此，不存在任何元素 l，使得 C' 中以 s_i 和 s_j 为指标的列存在差异，对角线元素除外。□。

4 置换德米坚科矩阵的识别算法

在本节中，我们介绍一种 O（n^4）算法，用于解决 Demidenko 矩阵的识别问题。伪代码见算法 1，其中涉及下文介绍的 CheckCandidateDP 过程。其基本思想是利用第 3 节所述的 Demidenko 矩阵与 Anti-Robinson 矩阵之间的关系，并使用文献中已知的某种置换 Anti-Robinson 矩阵识别算法，参见 [19, 21] 及其中的参考文献。更具体地说，我们使用 Pr´ea 和 Fortin [21] 提出的 O（n^2）算法来识别 $n \times n$ Anti-Robinson 矩阵。下文中

该算法记为 ALG。 ALG 以一个对称矩阵及其大小作为输入，并输出一个逻辑变量：如果输入是置换 Anti-Robinson 矩阵，则其值为真，否则为假。 ALG 还具有第二个输出；当逻辑输出为真时，该输出是输入矩阵的一个 Anti-Robinson 排列。如果逻辑输出为假，则 ALG 的第二个输出无关紧要，可以是任意值。注意，当输入规模为 $O(n^2)$ 时， ALG 的时间复杂度 $O(n^2)$ 已是可能达到的最佳结果。

算法 1 置换德米坚科矩阵的识别算法

1: **过程** RECOGNPD(C,n)
2: **对** $p = 1$ 至 n **执行**
3: **对** $q = 1$ 至 $n, p \neq q$ **执行**
4: 集合 $r = 0$, $\pi(1) = p$, $\pi(n) = q$.
5: 令 $\pi(i) = 0$ 对 $i \in \{2, 3, \ldots, n-1\}$。
6: **对于** $i = 1$ 至 n **执行**
7: **对于** $i = 1$ 至 n **执行**
8: 集合 $c'_{ij} = c_{ij} - c_{pj} - c_{ip}$.
9: **结束循环**
10: **结束循环**
11: (IsDPCandidate,π)=CheckCandidateDP($C' = (c'_{ij})$ n, r, π)
12: **如果**IsDPCandidate=真 **则**
13: **如果**$(C')^{\pi}$ 满足条件 (3) **则**
14: 返回“C 是一个置换 Demidenko 矩阵，其 Demidenko 置换为 mutation π”，并停止。
15: **结束 if**
16: **结束 if**
17: **结束循环**
18: **结束循环**
19: 返回“C 不是置换 Demidenko 矩阵”
20: **结束过程**

算法 1 将对称的 $n \times n$ 矩阵 C 及其阶数 n 作为输入，并检查是否存在 Demidenko 排列 τ ，使得对于 C ， $\tau(1) = p$ 以及 $\tau(n) = q$，其中每一对固定的指标 p、$q \in I$ 均满足：$= \{1, 2, \ldots, n\}$、 $p \neq q$。这是通过调用过程 CheckCandidateDP（见算法 1 的第 11 行）完成的，该过程有四个输入。前两个输入是矩阵 C' （按照引理 3.5 中的规定计算）及其阶数 n。第三个输入是部分排列 π。CheckCandidateDP 检查π 是否可以补全为 C的候选 Demidenko 排列。在第一次调用 CheckCandidateDP 时，通过设置 $\pi(1) = p$、 $\pi(n) = q$ 以及 $\pi(i) = 0$，将 π 初始化，其中$i \notin \{2, \ldots, n-1\}$。因此，在初始化时，部分排列 π中只有两个元素被固定。在递归调用 CheckCandidateDP 的过程中， π 中的更多元素会被固定。实际上，CheckCandidateDP 的第三个输入是 π 中除 $\pi(n)$之外已固定元素的数量 r 。CheckCandidateDP(C', n, r, π) 返回两个值：逻辑值 IsDPCandidate；当可以将 π 补全为 C 的候选 Demidenko 排列时，该值为真，否则为假。第二个输出是 π；如果返回 IsDPCandidate 真，则它是矩阵 C' （以及 C）的候选 Demidenko 排列；如果返回 IsDPCandidate 假，则它是一个（已废弃的）部

分排列。假设 CheckCandidateDP 返回 (真,π)。在这种情况下，引理 3.4 和 3.5 保证： π 是 C' 的 Demidenko 排列（因而也是 C的 Demidenko 排列），前提是存在一个 Demidenko 排列 C，将 1 和 n 分别映射到 p 和 q的当前值。因此，算法 1 检查 π 是否是 C' 的 Demidenko 排列；如果是，则输出相应消息并停止（见算法 1 的第 14 行）。在否定情况下，不存在将 1 映射到 p 、将 n 映射到 q 的 C 的 Demidenko 排列（参见引理 3.4 和 3.5）。随后，针对下一对 (p, q) 调用 CheckCandidateDP。如果在双重 for 循环结束时 IsDPCandidate 仍等于假，则不存在 C的 Demidenko 排列，算法 1 在第 19 行输出相应消息并停止。

过程 CheckCandidateDP 首先计算量 $m := \min\{\sum_{x=1}^{r} c_{i\pi(x)} - rc_{i\pi(n)} : i \in I'\}$和 $K := \{i \in I' : \sum_{x=1}^{r} c_{i\pi(x)} - rc_{i\pi(n)} = m\}$，其中 $I' = \{1, 2, \ldots, n\} \setminus \{\pi(i) : i \in \{1, 2, \ldots, r\} \cup \{n\}\}$ （如引理 3.4 中所述）。然后区分以下三种情况： $|K| = 1$、 $|K| = n - r - 1$ 和 $1 < |K| < n - r - 1$。

在情况 $|K| = 1$ $K = \{s\}$ 中，算法将 $\tau(r + 1) = s$ 设为相应值。在这种情况下，引理 3.4 意味着，对于每个将指标 $\{1, 2, \ldots, r, n\}$ 映射为不完整排列 π 所指定形式的 Demidenko 排列 τ，都必须满足 $\tau(r + 1) = s$。随后相应地更新 π和 r，并递归调用 CheckCandidateDP。

在 $|K| = n - r - 1$的情况下，应用 ALG 来检查按照引理 3.5(i) 定义的 $C'[K]$ 是否为反罗宾逊矩阵。若结果为否，即 ALG 返回假，则引理 3.5(i) 表明，不存在 Demidenko 排列能够针对 C 按照不完整排列 π 的指定方式映射指标 $\{1, 2, \ldots, r, n\}$。因此，CheckCandidateDP 返回假。若 ALG 返回真，则 $\psi \in \mathcal{S}_{n-r-1}$ 是相对于 $C'[K]$ 的 Anti-Robinson 排列。根据引理 3.5(i)，部分排列 π 随后可以补全为 C 的候选 Demidenko 排列，并且该排列在 K 上与 ψ 一致。CheckCandidateDP 返回真，并给出 C 的候选 Demidenko 排列 π。

最后，在 $1 < |K| < n - r - 1$的情况下，给定 $K = \{s_1, s_2, \ldots, s_{|K|}\}$ 和 $s_1 \leq s_2 \leq \ldots \leq s_{|K|}$，按照引理 3.5(ii) 的说明构造 $(|K| + 1) \times (|K| + 1)$ 矩阵 D（过程 CheckCandidateDP 的第 22 至 30 行）。接着，应用 ALG 来检查 D 是否为置换 Anti-Robinson 矩阵。若结果为否，引理 3.5(ii) 表明，不存在 Demidenko 排列能够针对 C 按照不完整排列 π 的指定方式映射指标 $\{1, 2, \ldots, r, n\}$。因此，CheckCandidateDP 返回假。若结果为是，CheckCandidateDP 将依据引理 3.5(ii) 的第二个陈述，固定 π 在更多 $|K|$ 指标处的取值（CheckCandidateDP 第 34 行）。随后，相应地更新 π和 r，并递归调用 CheckCandidateDP。

综上所述，我们得出结论：算法 1 能正确判断其输入是否为置换 Demidenko 矩阵；若是，则输出该输入矩阵的 Demidenko 排列。

1: 过程 $\textsc{CheckCandidateDP}(C, n, r, \pi)$ **2: 设** $I' := \{1, 2, \ldots, n\} \setminus \{\pi(i): i \in \{1, 2, \ldots, r\} \cup \{n\}\}$**. 3: 设** $m := \min\{\sum_{x=1}^{r} c_{i\pi(x)} - r c_{i\pi(n)}: i \in I'\}$**. 4: 设** $K := \{i \in I': \sum_{x=1}^{r} c_{i\pi(x)} - r c_{i\pi(n)} = m\}$**. 5: 如果** $|K| = 1$ **则 6: 设** $\pi(r+1) = x$ **对于** $x \in K$**. 7: 设** $r := r + 1$**. 8: (IsDPCandidate,**π**)=CheckCandidateDP(**C, n, r, π**) 9: 结束如果 10: 如果** $|K| = n - r - 1$ **则 11: 计算** $C[K]$ **对于** $K = \{s_1, s_2, \ldots, s_{|K|}\}$**. 12: (IsDPCandidate,**ψ**)=**$\mathrm{ALG}(C[K], |K|)$ ▷ **检查** $C[K]$ **是否为置换 Anti-Robinson 矩阵 13: 如果 IsDPCandidate=为真 then 14: 对于** $i = 1$ **到** $n - r - 1$ **执行 15: 设** $\pi(r + i) = s_{\psi(i)}$**. 16: 结束循环 17: 结束如果 18: 返回 (IsDPCandidate,**π**) 19: 结束如果 20: 如果** $1 < |K| < n - r - 1$ **则 21: 设** $M := 2\sum_{i=1} \sum_{j=1}^{n} |c_{ij}|$ ▷ **生成大量** M**22: 对于** $i = 1$ **到** $|K|$ **执行** ▷ **按照引理 3.5 构造矩阵** D **23: 对于** $j = 1$ **到** $|K|$ **执行 24: 设** $d_{ij} := c_{s_i, s_j}$**25: 结束循环 26: 结束循环 27: 对于** $i = 1$ **到** $|K|$ **执行 28:** $d_{i,|K|+1} := M \sum_{l \in I' \setminus K} c_{s_i l}$**,** $d_{|K|+1,i} := d_{i,|K|+1}$**29: 结束循环 30: 设** $d_{|K|+1,|K|+1} := M$**31: (IsDPCandidate,**ψ**)=**$\mathrm{ALG}(D, |K| + 1)$ ▷ **检查** D **是否为置换 Anti-Robinson 矩阵 32: 如果 IsDPCandidate=为真 then 33: 对于** $i = 1$ **到** $|K|$ **执行 34: 设** $\pi(r + i) = s_{\psi(i)}$**35: 结束循环 36: 设** $r := r + |K|$**37: (IsDPCandidate,**π**)=CheckCandidateDP(**C, n, r, π**) 38: 结束如果 39: 返回 (IsDPCandidate,**π**) 40: 结束如果 41: 结束过程**

算法的时间复杂度。我们注意到，对于每一对固定的指标 (p, q)，算法 1 都可以实现为在 $O(n^2)$ 时间内运行，因此总时间复杂度为 $O(n^4)$。具体而言，考虑固定的一对指标 (p, q) 所需的计算开销。计算 C' 显然需要 $O(n^2)$ 时间。CheckCandidateDP 包含两个主要操作：确定指标集合 K，以及识别一个经排列的 Anti-Robinson 矩阵（$C[K]$ 或 D）。确定 K 依赖于计算 CheckCandidateDP 第 3 行中的各个和。请注意，在 CheckCandidateDP 过程中，部分排列 π 会被赋予新值，但已有值不会改变。因此，对于 CheckCandidateDP 的所有递归调用，上述各个和均可增量计算，总共需要 $O(n^2)$ 时间。显然，每次调用 CheckCandidateDP 时，计算 m 和 K 都可以在 $O(n)$ 时间内完成。注意到 CheckCandidateDP 的递归调用所处理的 $\{1, 2 \ldots, n\}$ 的子集 K 两两不相交，由此可知，这类调用至多有 n 次。因此，计算 m 和 K 总共需要 $O(n^2)$ 时间。最后，考虑所有 ALG调用所产生的计算开销。由于 ALG 作用于 C' 的两两不相交子矩阵上，且 $\sum_{i=1}^{l} n_i = n$的运行时间为二次，因此总时间复杂度为 $O(n^2)$。

综上，我们得到以下定理。

定理4.1 算法 *1* 能正确解决 *Demidenko* 矩阵类的识别问题。其运行时间为 $O(n^4)$，其中 n 是输入矩阵的规模。

5 总结

本文提出了一种 $O(n^4)$ 算法，用于识别规模为 n 的置换德米坚科矩阵，从而解决了一个悬而未决数十年的问题。该算法填补了识别问题领域中一个显著的空白。事实上，Demidenko 矩阵类的许多子类的识别问题早已得到解决，例如 Kalmanson矩阵、Supnick 矩阵、Monge 矩阵和 Anti-Robinson 矩阵；然而，它们共同的超类，即 Demidenko 矩阵，识别问题仍未解决。

我们的算法在理论上基于 Demidenko 矩阵与 Anti-Robinson 矩阵之间的关系。它利用了已知的置换 Anti-Robinson 矩阵识别算法，尤其是 [21]。高效识别置换德米坚科矩阵，扩大了一些著名组合优化问题（如旅行商问题或路径旅行商问题）的多项式时间可解情形。

参考文献

[1] D.L. APPLEGATE, R.E. BIXBY, V. CHVATAL, W.J. COOK (2006)。旅行商问题。计算研究。普林斯顿大学出版社。[2] R.E. BURKARD, V. DEINEKO, R. VAN DAL, J.A.A. VAN DER VEEN AND G.J.WOEGINGER(1998)。TSP 的易求解特殊情形：综述。《*SIAM*评论》，**40(3)**，496–546。

[3] R.E. Burkard, B. Klinz and R. Rudolf (1996)。优化中的蒙日性质。离散应用数学，70(2)，95–161。[4] E. Çela, V. Deineko and G.J. Woeginger (2018)。具有对角结构系数矩阵的二次指派问题的新特殊情形。《欧洲运筹学杂志》，267，818–834。[5] E. Çela, V. Deineko and G.J. Woeginger (2022)。Demidenko 矩阵上的旅行商路径，《离散应用数学》，即将刊出。在线发布于 2021 年 12 月 10 日。https://doi.org/10.1016/j.dam.2021.11.019[6] V. Deineko and V.L. Filonenko (1979)。关于特殊结构矩阵的重构。Aktualnyje Problemy EVM i programmirovanije，第聂伯罗彼得罗夫斯克，DGU，（俄文）。[7] V. Deineko, B. Klinz, A. Tiskin and G.J. Woeginger(2014)。TSP 的四点条件：完整的复杂性分类。《离散优化》，14，147–159。[8] V. Deineko, R. Rudolf and G.J. Woeginger (1996)。关于置换后的 Supnick 矩阵和不完全蒙日矩阵的识别。《信息学学报》33，559–569。[9] V. Deineko, R. Rudolf and G.J. Woeginger (1998)。有时旅行很容易：主巡回问题，《SIAM离散数学杂志》，11(1)，81–93。[10] V.M. Demidenko (1976)。旅行商问题的一个特殊情形。白俄罗斯苏维埃社会主义共和国科学院通报，物理与数学科学辑，5，28–32，（俄文）。[11] K. Fujii, N. Ito, S. Kim, M. Kojima, Y. Shinano and K.-C. Toh(2021)，求解具有挑战性的大规模二次指派问题，ZIB报告 21-02，2021 年 1 月，https://arxiv.org/abs/2101.09629v1。[12] M.R. Garey and D.S. Johnson (1979)。计算机与难解性：NP完全性理论指南。Freeman，旧金山。[13] P.C. Gilmore, E.L. Lawler and D.B. Shmoys (1985)。易求解的特殊情形。 [17], 87中的第 4 章–143。[14] G. Gutin and A.P. Punnen(2002)。旅行商问题及其变体。Kluwer Academic Publishers。[15] S.N. Kabadi (2002)。TSP 的多项式时间可解情形。 [14],中的第 11 章，489–576。[16] K. Kalmanson (1975)，边凸回路与旅行商问题，《加拿大数学杂志》，27，1000–1010。[17] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys(1985)。旅行商问题。Wiley，奇切斯特，1985。[18] M. Laurent and M. Seminaroti (2015)。具有托普利兹结构的罗宾逊型矩阵上的二次指派问题易于求解。《运筹学快报》，43(1)，103–109。[19] M. Laurent and M. Seminaroti (2017)。相似性优先搜索：一种应用于罗宾逊型矩阵识别的新算法。《SIAM离散数学杂志》，31(3)，1765–1800。

[20] C.H. Papadimitriou and M. Yannakakis (1993). 距离为1和2的旅行商问题。运筹学数学 *18*, 1–11.[21] P. Préa and D. Fortin. (2014). 识别Robinson型差异度的最优算法。分类学杂志, 31, 1–35.[22] S. Polyakovskiy,
F.C.R. Spieksma and G.J. Woeginger(2013). Kalmanson矩阵中的三维匹配问题。*J. Comb. Optim.*, 26(1), 1–9.[23] W.S. Robinson (1951). 一种按年代对考古堆积进行排序的方法。*A*merican Antiquity, 16, 293–301.[24] S. Sahni and T. Gonzalez (1976). P完全近似问题。*J*ournal of the ACM, 23, 555–565.[25] F. Supnick (1957). 极值哈密顿线。*A*nnals of Math., 66, 179–201[26] V. Traub,
J. Vygen and R. Zenklusen (2020). 将路径旅行商问题归约为旅行商问题。第*52*届*ACM SIGACT*理论计算年度研讨会论文集（*STOC 2020*）, 14-27. 即将在《*SIAM*计算杂志》发表。（另见 arXiv:arXiv:1907.10376[cs.DM]）[27] R. Zenklusen (2019). 路径旅行商问题的1.5近似。第*30*届*ACM-SIAM*离散算法年度研讨会论文集（*SODA-2019*）, 1539–1549.（另见 arXiv:1805.04131v2[cs.DM]）