arXiv:2304.10661v1 [cs.CC] 20 Apr 2023

一个与旅行商问题相关的猜想

杨健

罗格斯大学商学院管理科学与信息系统系

新泽西州纽瓦克，邮编
07102
电子邮箱：jyang@business.rutgers.edu

2000年9月；修订于2002年5月、2021年10月

摘要

我们表明，求解某些组合优化问题的特定方法，可以理解为利用查询平面将问题实例空间划分为多个多面体，而这些多面体可能嵌入刻画该问题各种解的多面体之中。这一视角自然引出了一个称为易碎裂性（splinter-proneness）的性质，随后我们证明，该性质是导致相关问题具有计算困难性的原因。我们猜想，NP-等价旅行商问题（TSP）具有这一性质，因此在一定程度上难以求解。

关键词： 组合优化问题；复杂性；旅行商问题

1 引言

传统上，人们将组合优化问题（COP）视为从问题实例映射到其某个最优解的过程。例如，旅行商问题（TSP）可以表述为：在给定相关城市信息的情况下，寻找成本最低的巡回路线；参见 [6]。另一种观点或许有助于我们更好地理解，为什么针对某一特定组合优化问题的单一算法能够应对该问题可能存在的无穷多种实例，并始终得出正确答案。

与针对单个实例的可行解不同，组合优化问题（COP）的可行解可以视为一个问题范围内的概念。此外，实例与其最优解之间的关系，与其说是“对应于”，不如说是“属于”。例如，对于有 $n+1$ 个城市的旅行商问题（TSP），共有 $n!$ 个可行解，也就是访问这些城市的 $n!$ 种排列。一个实例仅仅表达城市之间两两距离的信息，它属于与其具有相同最优解的实例集合。对于给定的可行解 s，我们稍后将包含所有以 s 为最优解的实例的集合称为 s 的 S-集。

按照上述对组合优化问题的理解，其算法可以视为实例空间中一系列依赖于历史的除法。首先，它将实例空间分为两半，分别称为 $+1$ 半和 -1 半。接着，它将 $+1$ 半分为 $+1{+}1$ 半和 $+1$-1 半；同时，它将 -1 半分为 -1$+1$ 半和 -1-1 半。当这一过程准确地持续足够长时间，使由其除法历史产生的每个集合都完全位于某个 S-集内时，我们就认为算法解决了该问题。换言之，组合优化问题的算法就是一棵二叉树，其中每个节点都是一个实例集合，每次分支都将父节点对应的集合除法为两个集合，分别作为两个子节点；树根是整个实例空间，而每个叶节点都完全位于某个 S-集内。因此，给定一个实例，算法执行定义该实例所属叶节点的一系列除法，然后报告其 S-集包含该叶节点的可行解，作为该实例的最优解。

对于各种组合优化问题（我们稍后将其称为实数输入线性组合优化问题），我们进一步认识到，它们各自的 S集合在必要时经过适度外推后，不过是一个多面体。对于这些组合优化问题，观察和直觉促使我们推测，就高效求解而言，只需要采用简单算法，通过充当查询的超平面划分实例空间。因此，问题的 S集合最终会包含所有由查询平面产生的半空间构成的多面体，而其面上的所有区域都需要由

某些查询平面进行遍历。因此，求解一个实数输入线性组合优化问题的难度，与寻找能够相对于其 S集合进行适当定位的查询平面的难度相关。特别是，我们发现了一种易碎性性质，并证明具有该性质的实数输入线性组合优化问题无法由任何简单算法在多项式时间内求解。我们还猜想，旅行商问题具有易碎性。由于该问题不仅是 NP难的，而且是 NP易的 [2],，这一猜想为广为人知且令人困惑的猜想 $P \neq NP$ 提供了充分条件：如果只需要简单算法即可求解旅行商问题。

本文其余部分安排如下。我们在第 2 节定义实数输入线性组合优化问题和简单算法，并在第 3 节详细阐述如何用后者求解前者。第 4 节提出易碎性概念，并说明它作为问题困难性的充分条件。在第 5 节，我们提出关于旅行商问题的猜想，并讨论相关主题。第 6 节为结论。

2 个 RILCOP 和简单算法

考虑一个具有以下特征的组合优化问题。

(i) 给定某个规模参数 n，其实例空间可以表示为一个固定维度的有理数空间，并且存在一个有限的可行解集合；

(ii) 在可行解下，存在一个关于该实例的线性目标值函数；此外，组合优化问题旨在使目标最小化。

指派问题（指派问题）和旅行商问题都属于这类实数输入线性组合优化问题（实数输入线性组合优化问题）。为便于呈现，我们一直坚持采用（ii）的单目标版本。在不改变本质的情况下进行轻微推广，可以使RILCOP 涵盖最短路径问题（最短路径问题）和最小生成树问题（最小生成树）。

例如，规模为 3 的 TSP 可以将其每个实例表示为一个 $3 \cdot (3+1) = 12$-分量有理向量：

$$\mathbf{c} \equiv (c_{01}, c_{02}, c_{03}, c_{10}, c_{12}, c_{13}, c_{20}, c_{21}, c_{23}, c_{30}, c_{31}, c_{32})^T. \tag{1}$$

所有实例共有 $3! = 6$ 个可行解：

$$\begin{cases} (3,0) = \text{ tour } 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 0, \\ (3,1) = \text{ tour } 0 \longrightarrow 1 \longrightarrow 3 \longrightarrow 2 \longrightarrow 0, \\ (3,2) = \text{ tour } 0 \longrightarrow 2 \longrightarrow 1 \longrightarrow 3 \longrightarrow 0, \\ (3,3) = \text{ tour } 0 \longrightarrow 2 \longrightarrow 3 \longrightarrow 1 \longrightarrow 0, \\ (3,4) = \text{ tour } 0 \longrightarrow 3 \longrightarrow 1 \longrightarrow 2 \longrightarrow 0, \\ (3,5) = \text{ tour } 0 \longrightarrow 3 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0. \end{cases} \tag{2}$$

对于每个可行解 $(3, s)$，都对应一个线性目标函数 $(\mathbf{x}\text{tsp}(3, s))^T \cdot \mathbf{c}$。例如，$\mathbf{x}\text{tsp}(3, 0) \equiv (1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0)^T$ 因此，

$$(\mathbf{x}_{\text{tsp}}(3,0))^T \cdot \mathbf{c} = c_{01} + c_{12} + c_{23} + c_{30}. \tag{3}$$

对于大小为 $n \in \mathbb{N}$ 的 RILCOP $\mathcal{P}$，其中 $\mathbb{N} \equiv \{1, 2, ...\}$ 表示自然数空间，我们将其整个实例空间记为 $I_{\mathcal{P}}(n)$。后者就是 $\mathbb{Q}^{D_{\mathcal{P}}(n)}$ 本身，其中 $\mathbb{Q}$ 是有理数空间，而 $D_{\mathcal{P}}(n)$ 是关于 n 的多项式，或该空间内的一个多面体。例如，$I\text{tsp}(n) = \mathbb{Q}^{D\text{tsp}(n)}$，其中 $D\text{tsp}(n) = n \cdot (n+1)$。我们还将该问题的不同可行解的数量记为 $S_{\mathcal{P}}(n)$，并将它们记作 (n, s)，其中 $s = 0$、1、...、$S_{\mathcal{P}}(n) - 1$。给出与 (n, s) 对应的线性目标的 $\mathbb{Q}^{D_{\mathcal{P}}(n)}$ 向量记为 $\mathbf{x}_{\mathcal{P}}(n, s)$。给定一个可行解 (n, s)，我们将以 s 为其最优解之一的实例集合记为 $I^S_{\mathcal{P}}(n, s)$，其相当于

$$\left\{ \mathbf{c} \in I_{\mathcal{P}}(n) : \ (\mathbf{x}_{\mathcal{P}}(n,s) - \mathbf{x}_{\mathcal{P}}(n,s'))^T \cdot \mathbf{c} \leq 0 \quad \forall s' = 0, ..., s-1, s+1, ..., S_{\mathcal{P}}(n) - 1 \right\}. \tag{4}$$

根据 (2) 至 (4)，S-集 $I^S\text{tsp}(3,0)$ 将包含 (1) 中所有满足条件的 $\mathbf{c} \in \mathbb{Q}^{12}$。

$$\begin{cases} \Big(\mathbf{x}_{\text{tsp}}(3,0) - \mathbf{x}_{\text{tsp}}(3,1)\Big)^T \cdot \mathbf{c} = (c_{01} + c_{12} + c_{23} + c_{30}) - (c_{01} + c_{13} + c_{32} + c_{20}) \\ \qquad\qquad = c_{12} - c_{13} - c_{20} + c_{23} + c_{30} - c_{32} \leq 0, \\ \Big(\mathbf{x}_{\text{tsp}}(3,0) - \mathbf{x}_{\text{tsp}}(3,2)\Big)^T \cdot \mathbf{c} = (c_{01} + c_{12} + c_{23} + c_{30}) - (c_{02} + c_{21} + c_{13} + c_{30}) \\ \qquad\qquad = c_{01} - c_{02} + c_{12} - c_{13} - c_{21} + c_{23} \leq 0, \\ \Big(\mathbf{x}_{\text{tsp}}(3,0) - \mathbf{x}_{\text{tsp}}(3,3)\Big)^T \cdot \mathbf{c} = (c_{01} + c_{12} + c_{23} + c_{30}) - (c_{02} + c_{23} + c_{31} + c_{10}) \\ \qquad\qquad = c_{01} - c_{02} - c_{10} + c_{12} + c_{30} - c_{31} \leq 0, \\ \Big(\mathbf{x}_{\text{tsp}}(3,0) - \mathbf{x}_{\text{tsp}}(3,4)\Big)^T \cdot \mathbf{c} = (c_{01} + c_{12} + c_{23} + c_{30}) - (c_{03} + c_{31} + c_{12} + c_{20}) \\ \qquad\qquad = c_{01} - c_{03} - c_{20} + c_{23} + c_{30} - c_{31} \leq 0, \\ \Big(\mathbf{x}_{\text{tsp}}(3,0) - \mathbf{x}_{\text{tsp}}(3,5)\Big)^T \cdot \mathbf{c} = (c_{01} + c_{12} + c_{23} + c_{30}) - (c_{03} + c_{32} + c_{21} + c_{10}) \\ \qquad\qquad = c_{01} - c_{03} - c_{10} + c_{12} - c_{21} + c_{23} + c_{30} - c_{32} \leq 0. \end{cases}$$

对于旅行商问题，与许多其他实数输入线性组合优化问题一样，似乎没有必要进行以下操作以外的运算：将一个分量乘以一个常数有理数的乘法、将第一次运算所得的两个有理数相加、比较所得的任意有理数与零，以及不涉及实例的其他运算。在用于求解指派问题的匈牙利算法中，其他运算包括求解一个完整的无权最大匹配问题。在匈牙利算法的某个阶段，指示要解决哪个无权最大匹配问题的 0-1 信息确实依赖于实例；然而，求解过程本身并不涉及前三种运算。我们将仅采用上述四种运算的算法称为简单算法。

本质上，简单算法是在实例上始终执行线性运算，且从不通过非算术运算破坏实例的算法。用于最短路径问题和最小生成树的迪杰斯特拉算法 [1], 用于后一个问题的克鲁斯卡尔算法 [4], 用于指派问题的匈牙利算法 [5]都是简单算法的例子。

这里有一个颇具讽刺意味的反例。线性规划（LP）可以看作一种组合优化问题，但不是 RILCOP。此外，即使允许将比较对象设为任意非零有理数，单纯形法也不是简单算法。一个有 m 行、 n 列的标准型线性规划（$n \geq m$）可以用一个含有$n + mn + m$ 个分量的有理数向量表示，例如

$$\mathbf{e} \equiv (c_1, ..., c_n, a_{11}, ..., a_{1n}, ..., a_{m1}, ..., a_{mn}, b_1, ..., b_m)^T,$$

其中， c_j是通常所说的目标系数， a_{ij}是约束系数，b_i是右端项系数。令 $(n, m, 0)$、$(n, m, 1)$、...、$(n, m, n!/(m! \cdot (n-m)!) - 1)$ 表示所有 $n!/(m! \cdot (n-m)!)$个基，不考虑其可行性。我们能将线性规划建模为 RILCOP 的最接近方式，是将其在每个 (n, m, s)-可行解处的目标值写作

$$(\mathbf{x}_{\mathrm{lp}}(n, m, s, \mathbf{e}))^T \cdot \mathbf{e} = \sum_{j=1}^{n} x_{\mathrm{lp},j}(n, m, s, \mathbf{e}) \cdot c_j$$

其中，若可行，$\mathrm{xlp}(n, m, s, \mathrm{e}) \equiv (^x\mathrm{lp}_{,1}(n, m, s, \mathrm{e}) ..., ^x\mathrm{lp}_{\,n}(n, m, s, \mathrm{e}))^T$是在实例 e 下对应于基 (n, m, s) 的基本可行解；若 (n, m, s) 不是 e 的可行基，则为 $+\infty$。由于 $\mathrm{xlp}(n, m, s, \mathrm{e})$ 明显依赖于 e，LP 实际上不会是一个实数输入线性组合优化问题。单纯形法也不是简单算法。它有时会对两个操作数均为实例分量的数进行除法。对于实数输入线性组合优化问题，我们尚未见到有人尝试使用非简单算法求解。

3 通过简单 $\mathcal{A}$求解 RILCOP $\mathcal{P}$

在大小为 n 的 RILCOP $\mathcal{P}$ 上运行简单算法 $\mathcal{A}$ 时，$I_{\mathcal{P}}(n)$ 中的所有实例在 $\mathcal{A}$中的第一次比较之前，都会经历相同的运算序列。这次比较关乎是否存在某个 $\mathbf{q}_{\mathcal{A}}(n) \in \mathbb{Q}^{D_{\mathcal{P}}(n)}$，使得 $(\mathbf{q}_{\mathcal{A}}(n))^T \cdot \mathbf{c} \leq 0$。根据答案是肯定还是否定，这次首次比较会将 $I_{\mathcal{P}}(n)$ 分成两个多面体 $I_{\mathcal{P},\mathcal{A}}(n,+1)$ 和 $I_{\mathcal{P},\mathcal{A}}(n,-1)$。$I_{\mathcal{P},\mathcal{A}}(n,+1)$ 中的所有实例接下来也会经历相同的运算序列，直到下一次比较。这次比较关乎是否存在某个 $\mathbf{q}_{\mathcal{A}}(n,+1) \in \mathbb{Q}^{D_{\mathcal{P}}(n)}$，使得 $(\mathbf{q}_{\mathcal{A}}(n,+1))^T \cdot \mathbf{c} \leq 0$。同样，根据答案是肯定还是否定，这次比较会将 $I_{\mathcal{P},\mathcal{A}}(n,+1)$ 分成 $I_{\mathcal{P},\mathcal{A}}(n,+1,+1)$ 和 $I_{\mathcal{P},\mathcal{A}}(n,+1,-1)$。对于 $I_{\mathcal{P},\mathcal{A}}(n,-1)$ 也可作类似陈述；对于任意 $I_{\mathcal{P},\mathcal{A}}(n,\mathbf{b})$，其中 $\mathbf{b}$ 是由 $+1$ 和 -1 组成的二进制序列，这种过程会一直持续到 RILCOP 得到求解为止。当对于某个二进制序列 $\mathbf{b}$ 和 $s = 0, 1, ..., S_{\mathcal{P}}(n) - 1$，有 $I_{\mathcal{P},\mathcal{A}}(n,\mathbf{b}) \subseteq I^S_{\mathcal{P}}(n,s)$ 时，通过查询 $(\mathbf{q}_{\mathcal{A}}(n,b_1,...,b_{l-1}))^T \cdot \mathbf{c} \leq 0$ 是否成立，所得答案 b_l 足以检测出 $\mathbf{c} \in I^S_{\mathcal{P}}(n,s)$。

不失一般性，假设 $I_{\mathcal{P}}(n)$ 就是 $\mathbb{Q}^{D_{\mathcal{P}}(n)}$本身。那么，整个 RILCOP$\mathcal{P}$ 可以用这些序列表示。

$$(D_{\mathcal{P}}(n))_{n\in\mathbb{N}}, \quad (S_{\mathcal{P}}(n))_{n\in\mathbb{N}}, \quad \text{and} \quad \left((\mathbf{x}_{\mathcal{P}}(n,s))_{s=0,1,...,S_{\mathcal{P}}(n)-1}\right)_{n\in\mathbb{N}}, \tag{5}$$

其中每个 $\mathbf{x}_{\mathcal{P}}(n,s) \in \mathbb{Q}^{D_{\mathcal{P}}(n)}$。此外，所有上述多面体都可以表示为有助于定义它们的形如 $\mathbf{f}^T \cdot \mathbf{c} \leq 0$ 的不等式集合，其中 $\mathbf{f}$ 是 $\mathbb{Q}^{D_{\mathcal{P}}(n)}$ 中的向量。现在，每个 $I_{\mathcal{P}}(n)$ 都只是空集（），并且根据 (4)，每个$I^S_{\mathcal{P}}(n,s)$ 都只是一个（$S_{\mathcal{P}}(n) - 1$）元的 $D_{\mathcal{P}}(n)$ 维集合。

$$(\mathbf{x}_{\mathcal{P}}(n,s) - \mathbf{x}_{\mathcal{P}}(n,s'))_{s'=0,...,s-1,s+1,...,S_{\mathcal{P}}(n)-1}. \tag{6}$$

为了更精确地描述算法 $\mathcal{A}$，令 $\mathscr{B}_l$ 为所有 2^l 个长度为 l 的（$+1$, -1）值二进制字符串的集合。当对于任意 $\mathbf{b} \equiv (b_1, ..., b_l) \in \mathcal{B}$，其中 $l = 1, 2, ...$，其“父”字符串 $\mathbf{b} \ominus (b_l) \equiv (b_1, ..., b_{l-1})$ 以及“兄弟”字符串 $\mathbf{b} \ominus (b_l) \oplus (-b_l) \equiv (b_1, ..., b_{l-1}, -b_l)$ 也都属于 $\mathcal{B}$ 时，我们称字符串集合 $\mathcal{B}$ 具有树状结构。当对于任意 $n \in \mathbb{N}$，都存在一个具有树状结构的集合 $\mathcal{B}_{\mathcal{A}}(n) \subset \bigcup_{l=0}^{+\infty} \mathscr{B}_l$，使得对于任意字符串 $\mathbf{b} \in \mathcal{B}_{\mathcal{A}}(n)$，都存在一个查询 $\mathbf{q}_{\mathcal{A}}(n,\mathbf{b}) \in \mathbb{Q}^{D_{\mathcal{P}}}(n)$ 时，我们称简单算法 $\mathcal{A}$ 与通过（5）定义的 RILCOP $\mathcal{P}$ 相容。也就是说，一个与 $\mathcal{P}$ 相容的 $\mathcal{A}$ 本质上是

$$(\mathcal{B}_{\mathcal{A}}(n))_{n\in\mathbb{N}} \quad \text{and} \quad \left((\mathbf{q}_{\mathcal{A}}(n,\mathbf{b}))_{\mathbf{b}\in\mathcal{B}_{\mathcal{A}}(n)}\right)_{n\in\mathbb{N}},$$

其中每个 $\mathcal{B}_{\mathcal{A}}(n)$ 都是树状的，并且每个 $\mathbf{q}_{\mathcal{A}}(n, \mathbf{b}) \in \mathbb{Q}^{D_{\mathcal{P}}(n)}$。

给定一个 l-字符串 $\mathbf{b} \equiv (b_1, ..., b_l)$，我们用 $\mathbf{b}^{[l',l'']}$ 表示子串 $(b_m)_{m=l',...,l''}$。注意，在算法 $\mathcal{A}$ 中，大小参数为 n 时，与给定字符串 $\mathbf{b} \equiv (b_1, ..., b_l) \in \mathcal{B}_{\mathcal{A}}(n)$ 相关联的查询为 $\mathbf{q}_{\mathcal{A}}(n)$, $\mathbf{q}_{\mathcal{A}}(n, b_1)$, ..., $\mathbf{q}_{\mathcal{A}}(n, \mathbf{b})$，这些查询最终得到的两个多面体为 $(b_j \cdot \mathbf{q}_{\mathcal{A}}(n, \mathbf{b}^{[1,j-1]}))_{j=1,2,...,l} \oplus (+\mathbf{q}_{\mathcal{A}}(n, \mathbf{b}))$ 和 $(b_j \cdot \mathbf{q}_{\mathcal{A}}(n, \mathbf{b}^{[1,j-1]}))_{j=1,2,...,l} \oplus (-\mathbf{q}_{\mathcal{A}}(n, \mathbf{b}))$，其中 $\oplus$ 用于表示连接。重申一下我们的约定：前者表示 $\mathbb{Q}^{D_{\mathcal{P}}(n)}$ 内满足以下条件的所有向量 $\mathbf{c}$ 所构成的多面体

$$(b_1 \cdot \mathbf{q}_{\mathcal{A}}(n))^T \cdot \mathbf{c} \leq 0,$$

实际上是哪个 表示（$\mathbf{q}_{\mathcal{A}}(n))^T \cdot \mathbf{c} \leq 0$ 当 $b_1 = +1$ 时，以及（$\mathbf{q}_{\mathcal{A}}(n))^T \cdot \mathbf{c} \geq 0$ 当 $b_1 = -1$时 ，

$$(b_2 \cdot \mathbf{q}_{\mathcal{A}}(n, b_1))^T \cdot \mathbf{c} \leq 0, \quad ....., \quad \left(b_l \cdot \mathbf{q}_{\mathcal{A}}(n, \mathbf{b}_{[1,l-1]})\right)^T \cdot \mathbf{c} \leq 0,$$

最终为（$\mathbf{q}_{\mathcal{A}}(n,\mathbf{b}))^T \cdot \mathbf{c} \leq 0$。后一个多面体唯一的不同之处是最后一个线性不等式的符号相反。

给定一个树状二进制字符串集合 $\mathcal{B}$，我们可以将其叶子子集 $\mathcal{B}^0$ 定义为包含所有满足以下条件的$\mathbf{b}^0 \in \mathcal{B}$的集合：其子节点$\mathbf{b}^0 \oplus (+1)$ 和 $\mathbf{b}^0 \oplus (-1)$ 均不再属于 $\mathcal{B}$。事实上，后者可以理解为由 $\mathcal{B}^0$ 生成，因为每个$\mathbf{b} \in \mathcal{B}$ 都可以通过从某个$\mathbf{b}^0 \in \mathcal{B}^0$开始执行若干次 $\ominus$ 操作而获得。基于上述内容，我们可以认为，一个与 $\mathcal{P}$兼容的算法 $\mathcal{A}$ 能够求解 $\mathcal{P}$，当且仅当对于任意 $n \in \mathbb{N}$ 以及树状 $\mathcal{B}_{\mathcal{A}}(n)$ 的叶子子集 $(\mathcal{B}_{\mathcal{A}}(n))$ 中的任意$\mathbf{b} \equiv (b_1, ..., b_l)$，都存在某些 $s_+, s_- = 0, \mathbf{1}, ..., S_{\mathcal{P}}(n) - 1$，使得

$$\left(b_j \cdot \mathbf{q}_{\mathcal{A}}(n, \mathbf{b}_{[1,j-1]})\right)_{j=1,2,...,l} \oplus (\pm \mathbf{q}_{\mathcal{A}}(n, \mathbf{b})) \subseteq (\mathbf{x}_{\mathcal{P}}(n, s_\pm) - \mathbf{x}_{\mathcal{P}}(n, s'))_{s'=0,...,s_\pm-1,s_\pm+1,...,S_{\mathcal{P}}(n)-1}.$$

根据 Farkas 引理，上述内容可转化为存在 $2l \cdot (S_{\mathcal{P}}(n) - 1)$ 个正有理数 $\lambda^{\pm}_{s'1}$，....., $\lambda^{\pm}_{s',\ l-1}$，$\lambda^{\pm}_{s'l}$，对于 $s' = 0$，.....，$s_\pm - 1$，$s_\pm + 1$，.....，$S_{\mathcal{P}}(n) - 1$，使得对于 $+$ 或 - 以及每个这样的 $s' \neq s_\pm$，我们都有以下 $\mathbb{Q}^{D_{\mathcal{P}(n)}}$ 方程：

$$\mathbf{x}_{\mathcal{P}}(n, s_\pm) - \mathbf{x}_{\mathcal{P}}(n, s') = \sum_{j=1}^{l-1} \lambda^{\pm}_{s'j} b_j \cdot \mathbf{q}_{\mathcal{A}}(n, \mathbf{b}_{[1,j-1]}) \pm \lambda^{\pm}_{s'l} \cdot \mathbf{q}_{\mathcal{A}}(n, \mathbf{b}). \tag{7}$$

4 个面、易碎性与一个猜想

在 $I^S_{\mathcal{P}}(n,\ s)$ 的所有边界中，凡是包含满足以下条件的向量 $\mathbf{c}$的边界：对于某些固定的 $s' = 0$、.....、$s - 1$、$s + 1$、.....、$S_{\mathcal{P}}(n) - 1$，有 $(\mathbf{x}_{\mathcal{P}}(n,\ s) - \mathbf{x}_{\mathcal{P}}(n,\ s'))^T \cdot \mathbf{c} = 0$，其面便会显得突出。令$F_{\mathcal{P}}(n,\ s) \subseteq \{0, ..., s - 1, s + 1, ..., S_{\mathcal{P}}(n) - 1\}$为 $I^S_{\mathcal{P}}(n,\ s)$ 的面集。假设

$s' \in F_{\mathcal{P}}(n,\ s)$。那么，对于任意满足 $s''=0$，1，.....、$S_{\mathcal{P}}(n)-1$既非 s也非 s'的正有理数 $\lambda_{s''}$，以下情况不会发生：

$$\mathbf{x}_{\mathcal{P}}(n,s') = \sum_{s''=1 \text{ to } S_{\mathcal{P}}(n)-1,\ s''\neq s,\ s''\neq s'} \lambda_{s''}\cdot \mathbf{x}_{\mathcal{P}}(n,s'').$$

不同于（6），现在每个 $I_{\mathcal{P}}(n,\ s)$ 都可以按照.....的方式定义为面定义多面体

$$\left(\mathbf{x}_{\mathcal{P}}(n,s)-\mathbf{x}_{\mathcal{P}}(n,s')\right)_{s'\in F_{\mathcal{P}}(n,s)}.$$

对于查询平面 $\mathbf{q} \in \mathbb{Q}^d$ 与面定义多面体$I \equiv (\mathbf{y}^f)_{f\in F} \subseteq \mathbb{Q}^d$ 之间的关系，其中 d 是某个有限维度，F 是一个有限集，且每个$\mathbf{y}^f$ 都是 I的一个面，存在三种分类:

（spL）$\mathbf{q}$ 将 I 分成两半，却连它的任何一个面都没有刮到；更具体地说，这意味着不存在全正或全负的乘子组 $(\lambda_f)_{f\in F}$，使得$\mathbf{q} = \sum_{f\in F}\mathbf{y}^f$;

（sCp）$\mathbf{q}$ 刮到 I的一个面；更具体地说，这意味着存在唯一一个 $f\in F$以及某个 $\lambda_f \neq 0$，使得$\mathbf{q} = \lambda_f\cdot \mathbf{y}^f$;

（Rsp）$\mathbf{q}$ 从一侧远距离支撑 I，但不刮到它；更具体地说，这意味着存在全正或全负的乘子组 $(\lambda_f)_{f\in F}$，使得$\mathbf{q} = \sum_{f\in F}\lambda_f\cdot \mathbf{y}^f$，同时$\mathbf{q}$ 不与任何一个$\mathbf{y}^f$ 成正比；这也表明乘子组中至少有两个 λ_f非零。

这三个类别如图1所示。

上述概念也与平衡性概念相关。我们认为，当 $\sum_{k=1}^{d} q_k = 0$ 时，查询$\mathbf{q} \equiv (q_k)_{k=1,\ldots,d} \in \mathbb{Q}^d$ 是平衡的。若多面体 $I \equiv (\mathbf{y}^f)_{f\in F}$ 的每个定义查询 $\mathbf{y}^f$ 都是平衡的，则称该多面体是平衡的。

命题1 一个不平衡查询$\mathbf{q}$ 总会将一个平衡的多面体 $I \equiv (y_f)_{f\in F}$ 分裂成两个部分。

命题1的证明： 我们可以通过反证法证明这一点。假设一个不平衡查询$\mathbf{q}$没有分裂一个平衡的多面体 $I \equiv (y_f)_{f\in F}$。那么根据 (sCp) 和 (Rsp) 的定义，必定存在一个全正或全负的集合 $(\lambda_f)_{f\in F}$，使得$\mathbf{q} = \sum_{f\in F}\lambda_f\cdot \mathbf{y}^f$。如果我们对所有分量求和，就会得到

$$0 \neq \sum_{k=1}^{d} q_k = \sum_{k=1}^{d}\sum_{f\in F}\lambda_f\cdot y_{fk} = \sum_{f\in F}\lambda_f\cdot\sum_{k=1}^{d} y_{fk} = 0,$$

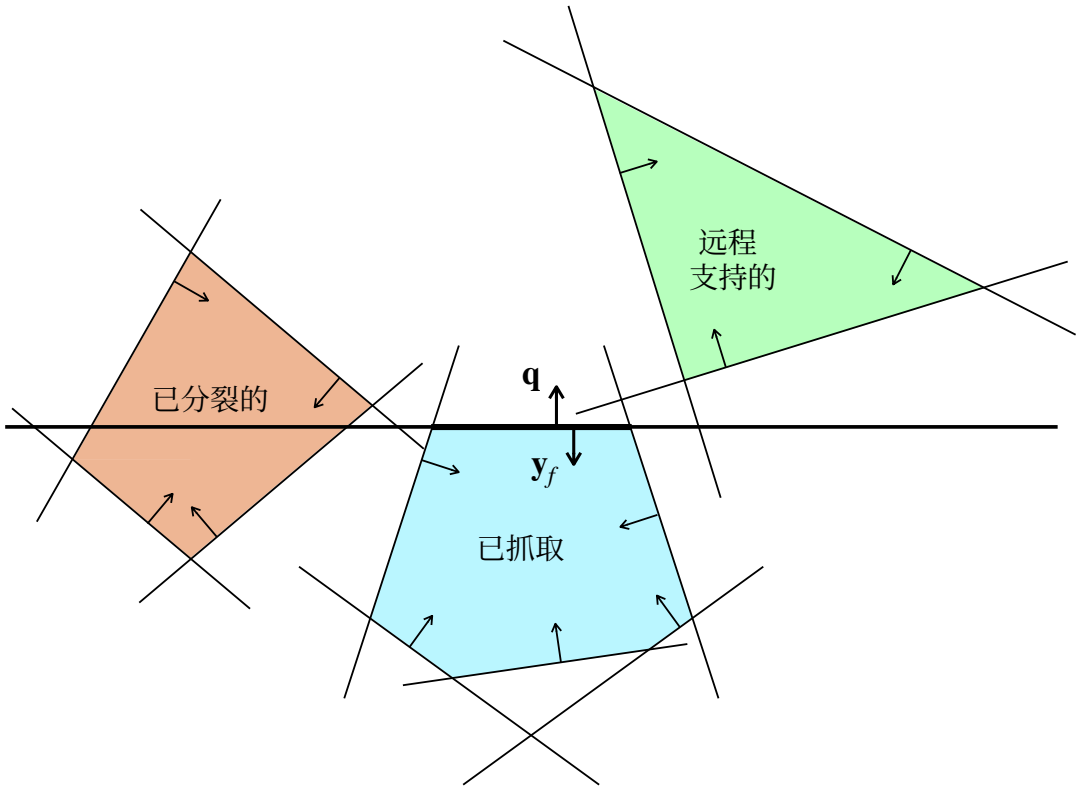

图1：三个类别的示意图

这显然构成矛盾。 ∎

当然，以式 (7) 的方式，使查询结果多面体进入每个 $I^S_{\mathcal{P}}(n,s)$，等价于所有面都已被刮除。

定义 1 当对于任意 $n \in \mathbb{N}$，所有 $I^S_{\mathcal{P}}(n,s)$ 的所有面点都将被 $\mathcal{A}$ 的查询平面刮取时，兼容的 $\mathcal{A}$ 将被视为已解决一个 *RILCOP* $\mathcal{P}$ 。

对于 $\mathbf{y} \neq \mathbf{0}$，我们用符号 $(\mathbf{y}|(\mathbf{y}_j)_{j\in J})$ 表示满足 $\mathbf{c}$ $\mathbb{Q}^d$个成员的集合，其中 $\mathbf{c}$满足 $\mathbf{y}^T \cdot \mathbf{c} = 0$且 $\mathbf{y}_j^T \cdot \mathbf{c} \leq 0$，其中 $j \in J$。对于 $\lambda \neq 0$，请注意 $(\lambda \cdot \mathbf{y}|(\mathbf{y}_j)_{i\in J})$ 表示 $\mathbb{Q}^{d-1}$中的一个多面体。现在更确切地说，定义 1 意味着，对于任意 $n \in \mathbb{N}$,$s = 0,1,...,S_{\mathcal{P}}(n) - 1$，以及 $s' \in F_{\mathcal{P}}(n,s)$，其 $(D_{\mathcal{P}}(n) - 1)$ 维面

$$\left(\mathbf{x}_{\mathcal{P}}(n,s) - \mathbf{x}_{\mathcal{P}}(n,s') \middle| \left(\mathbf{x}_{\mathcal{P}}(n,s) - \mathbf{x}_{\mathcal{P}}(n,s'')\right)_{s''\in F_{\mathcal{P}}(n,s)\setminus\{s'\}}\right) \tag{8}$$

始终会位于潜在面的并集中

$$\bigcup_{\mathbf{b}\equiv(b_1,\ldots,b_l)\in\mathcal{B}_{\mathcal{A}}(n)}\left(\mathbf{q}_{\mathcal{A}}(n,\mathbf{b})|\left(b_j\cdot\mathbf{q}_{\mathcal{A}}(n,\mathbf{b}_{[1,j-1]})\right)_{j=1,\ldots,l}\right). \tag{9}$$

对于任意长度为 l 的二进制串 $\mathbf{b}\in\mathcal{B}_{\mathcal{A}}(n)$，回想一下，$I_{\mathcal{P},\mathcal{A}}(n,\mathbf{b})$ 是将整个空间 $I_{\mathcal{P}}(n)\equiv\mathbb{Q}^{D_{\mathcal{P}}(n)}$ 与 l 半空间 $(b_j\cdot\mathbf{q}_{\mathcal{A}}(n,\mathbf{b}_{[1,j-1]}))$（其中 $j=1,...,l$）相交所得的乘积。正如 $I_{\mathcal{P}}(n)$ 可以理解为分解成若干个 $I^S_{\mathcal{P}}(n,s)$ 一样，由 $\mathbf{b}$ 确定的 $I_{\mathcal{P},\mathcal{A}}(n,\mathbf{b})$ 在某种意义上也可以分解成若干个 $I^S_{\mathcal{P},\mathcal{A}}(n,s,\mathbf{b})$，其中每个都是 $I^S_{\mathcal{P}}(n,s)$ 与上述 l 半空间相交所得的乘积。当然，这些 $I^S_{\mathcal{P},\mathcal{A}}(n,s,\mathbf{b})$ 中，有些维数已经低于 $D_{\mathcal{P}}(n)-1$，甚至为空。然而，剩余的满维部分仍可能数量可观。这个数量可能不会比初始的 $S_{\mathcal{P}}(n)$ 少多少。这样一来，算法 $\mathcal{A}$ 即使只针对 $I_{\mathcal{P},\mathcal{A}}(n,\mathbf{b})$，也仍会面临与一开始处理整个 $I_{\mathcal{P}}(n)$ 时基本同样困难的问题。

此外，结合三分类以及 (8) 到 (9) 的可解性判据，我们得出了以下性质；我们认为，这一性质是导致 RILCOP 难以由任何简单算法处理的关键。由于该性质与任何特定的 $\mathcal{A}$ 无关，我们需要更新使用二进制字符串的符号表示。给定任意非零的 $D_{\mathcal{P}}(n)$ 维序列 $(\mathbf{q}_1,\ldots\ldots,\mathbf{q}_l)$，令 $I_{\mathcal{P},\,Q}(n,\,\mathbf{q}_1,\,\ldots\ldots,\,\mathbf{q}_l)$ 表示将 $I_{\mathcal{P}}(n)$ 与 l 个半空间 $(\mathbf{q}_j)$（其中 $j=1,\ldots\ldots,l$）相交所得的积。$I^S_{\mathcal{P},\,Q}(n,\,s,\,\mathbf{q}_1,\,\ldots\ldots,\,\mathbf{q}_l)$ 也按类似方式定义。

定义 *2* 若一个 *RILCOP* $\mathcal{P}$ 中有指数级数量的 $I^S_{\mathcal{P}}(n,s)$ 完全维满，则称其具有碎片倾向；更重要的是，只要在经过多项式数量的 $p(n)$ 次查询 $q_1,...,q_{p(n)}$ 后，$I^S_{\mathcal{P},Q}(n,s,q_1,...,q_{p(n)})$ 中完全维满者的数量仍为某个指数级 $e(n)$，那么，为了对这些 $e(n)$ 多面体进行另一次查询，若使用的 q 与此前出现过的 q_j 成比例，就会遇到某些 $e(n)-q(n)$ 碎片，而 $q(n)$ 仅为多项式级。

一个易产生碎片化的难题 $\mathcal{P}$，其解所诱导的多面体集合在接受平面查询时，基本上大多数都会落入类别 (spL)；只有多项式数量的此类集合会归入类别 (sCp) 和 (Rsp)。更准确地说，即使经过与多项式数量的半空间求交，所有有助于定义该问题的指数级数量的解集的指数级数量的不等式，也只会使其中多项式数量的集合明确地位于给定查询平面的某一侧，表现为 (sCp) 或 (Rsp) 形式。而绝大多数集合则会以 (spL) 形式同时出现在查询平面的两侧。

定理 1 易产生碎片的 *RILCOP* $\mathcal{P}$ 将无法由任何与 $\mathcal{P}$ 兼容的简单算法 $\mathcal{A}$在多项式时间内求解。

定理 1 的证明： 在进行任何查询之前， $\mathcal{P}$ 肯定有指数级 $e_0(n) \leq S_{\mathcal{P}}(n)$ 个全维解集尚未被完全刮除。根据定义 1，该问题尚未解决。

假设经过多项式 $p(n)$ 次查询后，仍然存在指数级$e_{p(n)}(n) \equiv e_0(n) - q_1(n) - \cdots - q_{p(n)}(n)$ 个全维集合，它们是将所有 $\mathcal{P}$ 的 $e_0(n)$ 个解集与某些 $p(n)$ 个半空间相交后得到的；这些半空间构成了由过去所有查询诱导的 $2^{p(n)}$ 个潜在半空间流之一，其中$q_1(\cdot)$ ...，$q_{p(n)}(\cdot)$ 均为多项式；此外，这些集合中没有一个被完全刮除。注意$q(\cdot) \equiv q_1(\cdot) + \cdots + q_{p(\cdot)}(\cdot)$ 仍然是多项式。

考虑任意一个不与过去任何$p(n)$ 次查询成比例的第 $(p(n)+1)$ 次查询。由于 $\mathcal{P}$ 具有定义 2 所述的易碎裂性，因此与查询平面具有 (sCp) 或 (Rsp) 关系的 $e_{p(n)}(n)$ 个集合的数量为某个多项式 $q^0(n)$；其余每个 $e_{p(n)}(n) - q^0(n)$ 个集合都会被分成两部分。任选查询平面的一侧，某个多项式 $q_{p(n)+1}(n)$ 数量的 (sCp)-和 (Rsp)-集合会落在该侧，而其余 $q^0(n) - q_{p(n)+1}(n)$ 个则落在另一侧，其中 $q_{p(n)+1}(n) \leq q^0(n)$；与此同时，任一侧都会有 $e_{p(n)} - q^0(n)$ 个由 (spL) 产生的集合。在有 $q^0(n) - q_{p(n)+1}(n)$ 个 (sCp)-和 (Rsp)-集合的一侧，查询后的集合总数将为$e_{p(n)} - q^0(n) + q^0(n) - q_{p(n)+1}(n) = e_{p(n)}(n) - q_{p(n)+1}(n)$。

也就是说，通过将 $\mathcal{P}$的解集与 $p(n)+1$ 个半空间相交，可以推导出指数级$e_{p(n)+1}(n) \equiv e_{p(n)} - q_{p(n)+1}(n) \equiv e_0(n) - q_1(n) - \cdots - q_{p(n)}(n) - q_{p(n)+1}(n)$个全维集合；这些半空间构成了由所有 $p(n)+1$ 次先前查询所诱导的 $2^{p(n)+1}$ 种潜在半空间流之一；此外，这些集合中没有一个被完全抓取。请注意， $q'(\cdot) \equiv q(\cdot) + q_{p(\cdot)+1}(\cdot) \equiv q_1(\cdot) + \cdots + q_{p(\cdot)+1}(\cdot) + q_{p(\cdot)+1}(\cdot)$仍将是多项式级的。

从上述归纳法可知，在经过多项式级 $p(n)$次查询后，对于某个多项式级 $q(\cdot)$，仍会存在指数级 $e_0(n) - q(n)$个全维且尚未被完全抓取的集合；这些集合是通过将 $\mathcal{P}$原有的 $e_0(n)$ 个解集与 $p(n)$个半空间相交而得到的，而这些半空间构成了由 $p(n)$次查询所诱导的 $2^{p(n)}$ 种潜在半空间流之一。根据定义 1，经过这 $p(n)$次查询后， $\mathcal{P}$ 仍未被解决。

注意，与先前任一 $p(n)$ 次查询对齐的面都已被刮取，因此让第 $(p(n)+1)$ 次查询重复先前的查询不会带来额外的刮取。由于 $p(\cdot)$ 是任意的，这意味着 $\mathcal{P}$ 不可在多项式时间内求解。 ∎

鉴于定理 1，可在多项式时间内求解的指派问题必定不是易碎的。

5 旅行商问题

对于 TSP，回想一下，对每个 $n \in \mathbb{N}$，都有 $D\mathrm{tsp}(n) = n \cdot (n+1)$ 且 $^{S}\mathrm{tsp}(n) = n!$ 我们可以将 $\{1, 2, ..., n\}$ 的 $n!$ 个排列命名为 $\bar{\pi}(n, s)$，其中 $s = 0$ 取 $1, ..., n! - 1$，且每个 $\bar{\pi}(n, s) \equiv (\bar{\pi}(n, s, m))_{m=1}$，其中 n 取 2,...；此外，我们可以令 $\bar{\pi}(n, 0)$ 为恒等排列，其中每个 $\bar{\pi}(n, 0, m) = m$；另外，也可以令当 $s = 0$ 取 $1, ..., n! - 2$ 时，$\bar{\pi}(n, s)$ 按字典序排在 $\bar{\pi}(n, s+1)$ 之前，其含义是：仅当存在某个 $m = 2, ..., n$ 使得 $\bar{\pi}(n, s, m) \geq \bar{\pi}(n, s+1, m) + 1$ 时，才有某个 $m' = 1, ..., m-1$ 使得 $\bar{\pi}(n, s, m') \leq \bar{\pi}(n, s+1, m') - 1$。将 $\mathbb{Q}^{n\cdot(n+1)}$ 中的每个向量 y 理解为 $(y_{01}, y_{02}, ..., y_{0n}, y_{10}, y_{12}, ..., y_{1n}, ..., y_{n0}, y_{n1}, ..., y_{n,n-1})^T$ 后，我们可以令每个 $\mathrm{xtsp}(n, s)$ 为这样一个向量：其 $(0, \bar{\pi}(n, s, 1))$, $(\bar{\pi}(n, s, 1)\ \bar{\pi}(n, s, 2))$, ..., $(\bar{\pi}(n, s, n-1)\ \bar{\pi}(n, s, n))$, 以及 $(\bar{\pi}(n, s, n), 0)$ 分量为 1，其余分量为 0。

命题 2 对于旅行商问题，每个解集 $I^{S}tsp(n, s)$ 都是满维的，并且每个查询 $\mathbf{x}tsp(n, s) - \mathbf{x}tsp(n, s')$ 都会为 $I^{S}tsp(n, s)$ 生成一个面。

命题 2 的证明： 由于对称性，不妨设 $s = 0$。对于第一个断言，考虑实例 $\mathbf{c} \equiv (c_{01}, c_{02}, ..., c_{0n}, c_{10}, c_{12}, ..., c_{1n}, ..., c_{n0}, c_{n1}, ..., c_{n,n-1})^T \in \mathbb{Q}^{n\cdot(n+1)}$ 其中 $c_{01} = c_{12} = c_{23} = \cdots = c_{n-1,n} = c_{n0} = 0$，而其他每个 $c_{ij} = 1$。显然，它满足

$$\Big(\mathbf{x}_{\mathrm{tsp}}(n, 0) - \mathbf{x}_{\mathrm{tsp}}(n, s)\Big)^T \cdot \mathbf{c} < 0, \qquad \forall s = 1, 2, ..., n! - 1.$$

对于第二个断言，考虑任意的 $s = 1$、2、.....、$n! - 1$。现在令 $\mathbf{c} \in \mathbb{Q}^{n\cdot(n+1)}$ 满足 $c_{0,\bar{\pi}(n,s',1)} = c_{\bar{\pi}(n,s',1)\bar{\pi}(n,s',2)} = \cdots = c_{\bar{\pi}(n,s',n-1)\bar{\pi}(n,s',n)} = c_{\bar{\pi}(n,s',n)0} = 0$ 对于 $s' = 0$ 和 $s' = s$ 均成立，而对于其他任意 $c_{ij} = 1$。显然，

$$\Big(\mathbf{x}_{\mathrm{tsp}}(n, 0) - \mathbf{x}_{\mathrm{tsp}}(n, s)\Big)^T \cdot \mathbf{c} = 0 \text{ and yet}$$
$$\text{for any } s' \text{ other than 0 or the chosen } s, \Big(\mathbf{x}_{\mathrm{tsp}}(n, 0) - \mathbf{x}_{\mathrm{tsp}}(n, s')\Big)^T \cdot \mathbf{c} < 0.$$

只要该等式仍然成立，即使每个 c_{ij} 都受到最多达 $1/(2n+2)$ 的扰动，上述不等式仍可保持。

■

所有 $I^{S}\mathrm{tsp}(n, s)$ 也都是平衡的。事实上，还可以进一步说明：每个 $\mathbf{x}\mathrm{tsp}(n, s) - \mathbf{x}\mathrm{tsp}(n, s')$ 的 $n \cdot (n+1)$ 分量中，$+1$ 和 -1 的数量相等，其余位置均为 0。表 1 给出了一个例子。

表1：可构成多面体的查询示例

(i,j)	x旅行商问题$(n,0)$ –x旅行商问题$(n,1)$	$\cdots\cdots$	xtsp$(n,0)$ –x旅行商问题$(n,n!-1)$	$\mathbf{q}_1$	$\cdots$	$\mathbf{q}^{p(n)}$
$(0,1)$	$+1-1=0$	$\cdots\cdots$	$+1$	$q_{1,01}$	$\cdots$	$q_{p(n),01}$
$(0,2)$	0	$\cdots\cdots$	0	$q_{1,02}$	$\cdots$	$q_{p(n),02}$
$\cdots$	$\cdots$	$\cdots\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$(0,n)$	0	$\cdots\cdots$	-1	$q_{1,0n}$	$\cdots$	$q_{p(n),0n}$
$(1,0)$	0	$\cdots\cdots$	-1	$q_{1,10}$	$\cdots$	$q_{p(n),10}$
$(1,2)$	$+1-1=0$	$\cdots\cdots$	$+1$	$q_{1,12}$	$\cdots$	$q_{p(n),12}$
$\cdots$	$\cdots$	$\cdots\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$(1,n)$	0	$\cdots\cdots$	0	$q_{1,1n}$	$\cdots$	$q_{p(n),1n}$
$\cdots$	$\cdots$	$\cdots\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$(n,0)$	$+1$	$\cdots\cdots$	$+1$	$q_{1,n0}$	$\cdots$	$q_{p(n),n0}$
$(n,1)$	0	$\cdots\cdots$	0	$q_{1,n1}$	$\cdots$	$q_{p(n),n1}$
$\cdots$	$\cdots$	$\cdots\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$(n,n-1)$	-1	$\cdots\cdots$	-1	$q_{1,n,n-1}$	$\cdots$	$q_{p(n),n,n-1}$

指派问题也是如此。鉴于命题1，匈牙利算法只提出平衡的查询，大概不会令人意外。指派问题的目标是 $\sum_{i=1}^{n} c_{i\pi(i)}$；为与之相一致，旅行商问题的目标可以写成 $\sum_{i=0}^{n} c_{0,\ \pi((\pi^{-1}(n)+1)}$ 模 $(n+1)$，其中 $\pi(0)$按0处理。指派问题比旅行商问题容易得多的一个原因，或许在于：不同于覆盖大范围平衡向量的（$\mathbf{xap}(n,\ 0)\ -\mathbf{xap}(n,\ s)$），（$\mathbf{xtsp}(n,\ 0)\ -\mathbf{xtsp}(n,\ s)$）大概会遗漏许多角点。我们以定理1为基础，提出这个措辞更精确、也可能更具可操作性的猜想。

猜想1 按照定义*2*的含义，旅行商问题容易产生碎片。

猜想 1 的不完整证明： 首先，根据命题 2，显然每个 $I^S\mathbf{tsp}(n,\ s)$ 都是满维的。因此，我们可以放心地令 $e_0(n)$ 为指数级 $n!$ 。

下一步应尝试证明：对于任意多项式 $p(\cdot)$，都存在另一个多项式 $q(\cdot)$，使得对于任意 $n \in \mathbb{N}$、任意查询 $\mathbf{q}_1$ 、 ..... 、 $\mathbf{q}_{p(n)} \in \mathbb{Q}^{n\cdot(n+1)}$ 以及任意其他查询 $\mathbf{q} \in \mathbb{Q}^{n\cdot(n+1)}$，以下结论成立：

在与某个 $\mathbf{c}(s) \in \mathbb{Q}^{n\cdot(n+1)}$ 一同满足条件的 s 中，几乎全部（至多除外 $q(n)$ 个）

$$\Big(\mathbf{x}_{\mathrm{tsp}}(n,s) - \mathbf{x}_{\mathrm{tsp}}(n,s')\Big)^T \cdot \mathbf{c}(s) < 0, \qquad \forall s' = 0, ..., s-1, s+1, ..., n!-1,$$

并且，对于任意 $m = 1$、.....、 $p(n)$，$\mathbf{q}_m^T \cdot \mathbf{c}(s) < 0$ 都会贡献出某些 $\mathbf{c}_{+1}(s)$、$\mathbf{c}_{-1}(s) \in \mathbb{Q}^{n\cdot(n+1)}$，使得二者都满足 $\mathbf{c}(s)$ 的不等式，尽管不一定是严格满足；然而，

$$\mathbf{q}^T \cdot \mathbf{c}_{+1}(s) < 0, \qquad \text{and yet} \qquad \mathbf{q}^T \cdot \mathbf{c}_{-1}(s) > 0.$$

目前，我们还没有更多线索来应对这项艰巨的任务！ ∎

如果猜想 1 成立，我们就可以根据定理 1 推导出：按照定义 1 的含义，旅行商问题无法在多项式时间内求解。由于旅行商问题是一个 NP等价问题， [2][3], 只要证明 $P \neq NP$ 不存在能够求解旅行商问题的多项式时间算法即可。旅行商问题属于那些对简单算法极其重要的实数输入线性组合优化问题，因此，即使只证明这一较弱的性质，也似乎已为最终证明 $P - NP$猜想迈出了很大一步。

6 结语

我们已说明了我们对简单算法如何求解某些实数输入线性组合优化问题的理解。由此产生了与.....密切相关的易分裂性概念。

时间复杂度问题。关于 TSP 具有分裂倾向的猜想目前尚未得到证明。由于它是一个 NP 等价问题，或者是最容易的 NP-困难问题之一，因此这一潜在结论可能会对 $P \neq NP$ 问题产生相当深远的影响。

参考文献

[1] Dijkstra, E.W. 1959年. 关于与图相关的两个问题的注记。《数值数学》，**1**，第269-271页。[2] Garey, M.R. 和 D.S. Johnson. 1979年。《计算机与难解性：理论指南 NP-完全性。W.H. Freeman & Company，纽约。[3] Karp, R.M. 1979年。组合问题之间的可归约性。载于 R.E. Miller 和 J.W. Thatcher（编），《计算机计算的复杂性》，第85-103页。Plenum Press，纽约。[4] Kruskal, J.B.,Jr. 1956年。关于图的最短生成子树与旅行商问题。《美国数学会会刊》，**7**，第48-50页。[5] Kuhn, H.W. 1955年。指派问题的匈牙利算法。《海军研究后勤季刊》，**2**，第83-97页。[6] Lawler, E.L.、J.K. Lenstra、A.H.G. Rinnooy Kan 和 D.R. Shmoys。1985年。（编），旅行商问题。John Wiley & Sons，纽约。