《Lean 4 Machine-Verified Proof of P = NP via the Pedigree Polytope Membership Problem》研究报告

摘要

本文研究对象是 T. S. Arthanari 于 2026 年 6 月上传至 arXiv 的论文《Lean 4 Machine-Verified Proof of P = NP via the Pedigree Polytope Membership Problem》（arXiv:2606.03194v1）。论文宣称通过谱系多面体成员问题（Membership Problem for Pedigree Polytope，M3P）的强多项式算法证明对称旅行商问题（STSP）可在多项式时间内求解，进而得到 P=NP；同时宣称主要证明链已在 Lean 4/Mathlib4 中完成机器验证，且没有未解决的 `sorry`。

本报告首先重建论文的数学路线：用“谱系”编码逐阶段插入形成的 Hamilton 回路，以谱系特征向量的凸包作为研究对象，递归构造分层网络、禁弧运输问题和多商品流问题，并试图以多商品流达到最大值作为谱系多面体成员性的充要条件。随后，本报告分别审查论文中的数学推理、算法复杂度论证以及公开 Lean 4 工程的实际形式化内容。

审查得到的核心结论是：该论文没有建立其所宣称的 P=NP 结论，公开 Lean 工程也没有机器验证标准复杂度理论意义上的 P=NP。决定性原因包括：第一，把 MCF 判定转化为 M3P 判定所必需的“必要性”方向并未完成 Lean 证明，而论文又错误地声称只需要充分性即可推出 M3P∈P；第二，Lean 中的 `MCFFeasible` 结构直接携带了一个凸组合见证，所谓“充分性”证明并没有从流解独立构造目标点的凸组合；第三，P、NP、多项式时间、归约、STSP 属于 P 等概念没有按标准语义形式化，而是被声明为不透明的 `Prop` 或公理；第四，论文中的多面体 `A_n` 在核心文件中被定义成单元素类型 `Unit`；第五，名为 `M3P_in_P` 的 Lean 定理实际只证明一个刚性谱系列表长度上界，并没有证明任何算法存在性、正确性或运行时间；第六，从 M3P 到分离、优化、STSP 和 P=NP 的多个关键步骤被直接写成公理，其中若干公理已经包含了本应由论文证明的实质结论。

因此，Lean 能够成功编译最多说明：在作者选定的公理、占位类型和弱化后的定理陈述下，Lean 内核确认了相应形式命题可被推出。它不说明这些形式命题忠实表达了论文中的数学对象，也不说明标准意义上的 P=NP 已被证明。论文中的谱系表示、逐阶段插入模型、流网络构想和某些局部组合性质仍可能具有独立研究价值，但必须与其未经支持的复杂度结论严格区分。

一、研究对象、方法与评价标准

1. 研究对象

论文的中心主张可以压缩为以下推理链：

$$\mathrm{MCF}(n-1)\text{ 可强多项式求解} \Longrightarrow \mathrm{M3P} \in P \Longrightarrow \mathrm{STSP} \in P \Longrightarrow P = NP.$$

其中：

1. `P_n` 是所有规模为 n 的谱系的特征向量集合；
2. $\mathrm{conv}(P_n)$ 是谱系多面体；
3. M3P 是判定给定有理点 X 是否属于 $\mathrm{conv}(P_n)$；
4. MCF 是论文为成员判定构造的多商品流线性规划；
5. STSP 是对称旅行商问题。

论文的特殊之处不只在于声称解决 P vs NP，还在于强调该结论已经通过 Lean 4 验证。因此，审查必须同时回答两个不同问题：

1. 论文中的非形式化数学论证是否足以推出 P=NP?
2. Lean 工程实际证明的形式命题，是否忠实对应论文声称的数学结论?

这两个问题不能混为一谈。一个 Lean 工程可能无 `sorry` 地证明某个命题，但如果命题定义过弱、关键事实被当作公理、或对象没有忠实建模，那么机器验证并不能替代数学论证。反过来，非形式化论证中存在尚未形式化的部分，也不自动表示该数学结论必然错误；它只表示“已完成机器验证”的宣传不成立。

2. 本报告采用的核验方法

本报告进行了以下工作：

1. 阅读论文全文，重点核对定义、定理 3.3、5.1、6.8、6.9、6.10、7.5、7.7、9.1—9.3 和第 10 节的机器验证声明。
2. 梳理论文从谱系、MI 形式、分层网络、FAT、MCF 到 P=NP 的完整依赖链。
3. 检查作者公开仓库 `TiruArt/Pedigree-Polytopes-Lean4` 当前版本的核心 Lean 文件。
4. 对比论文中展示的 Lean 定理陈述与仓库中的真实类型。
5. 搜索核心目录中的 `axiom` 、 `sorry` 、占位定义和弱化结论。
6. 区分“论文明确证明”“论文只给草图”“引用外部定理”“Lean 中声明为公理”“Lean 中真正构造证明”五类证据。

评价标准不是“结论是否令人惊讶”，而是：

1. 定义是否准确；
2. 前提是否充分；

3. 推理方向是否正确；
4. 算法是否同时具备正确性、完备性和复杂度界；
5. 形式化命题是否与自然语言主张一致；
6. 外部公理是否被准确陈述，而非把待证结论包装成外部定理。

二、论文试图解决什么问题

1. P 与 NP 的背景

复杂度类 P 包含可以由确定性算法在输入长度的多项式时间内判定的问题。NP 包含“是”实例具有多项式长度证书、且证书可在多项式时间内验证的问题。显然 $P \subseteq NP$，而 P 是否等于 NP 是理论计算机科学的核心开放问题。

若任意一个 NP 完全问题被证明属于 P，则所有 NP 问题都可以通过多项式时间归约转化到该问题并被多项式求解，从而得到 P=NP。STSP 的判定版本是经典 NP 完全问题，因此证明 STSP∈P 确实足以推出 P=NP。

论文没有直接构造一个通常意义上的 TSP 动态规划或组合算法，而是走多面体路线：

1. 用三角形插入序列编码一条旅行商回路；
2. 研究这些编码点的凸包；
3. 设计该凸包的成员判定算法；
4. 由成员判定获得分离；
5. 由分离获得线性优化；
6. 用特定线性目标优化得到最短 Hamilton 回路。

这是一条经典的组合优化思路。真正困难的部分不是最后“NP 完全问题属于 P 则 P=NP”，而是前面能否得到一个正确、完整且编码复杂度受控的多项式算法。

2. 为什么谱系多面体可能有吸引力

传统 TSP 多面体直接使用边变量，常见方法包括 subtour elimination、割平面和 branch-and-cut。论文改用三角形变量描述“把新城市插入当前回路的一条边”。

从三城市回路 $1-2-3-1$ 开始，每一步把城市 k 插入已有边 $\{i_k, j_k\}$，删除该边并添加 $\{i_k, k\}$ 与 $\{j_k, k\}$。经过 $k = 4, \ldots, n$ 的插入后得到一个包含所有城市的 Hamilton 回路。

这种表示具有递归层次：

1. 第 k 层只决定城市 k 插入哪条已有边；
2. 每个选择可表示成三角形 $\{i_k, j_k, k\}$；

3. 一个完整解由每层一个三角形组成；
4. 相邻层之间存在“生成关系”。

论文希望利用这种递归结构，把全局凸组合一致性转化为网络流一致性。这一研究动机本身是合理的：如果每一层的局部分布能够通过规模受控的网络连接，并且这些局部连接恰好刻画全局谱系分布，就可能得到有效成员算法。

三、谱系、MI 形式与多面体

1. 谱系的定义

论文把谱系写成：

$$P = (\{1,2,3\}, \{i_4, j_4, 4\}, \dots, \{i_n, j_n, n\}).$$

它满足两个条件：

1. 每个后续三角形都有一个生成元，即被插入的公共边此前已经由结构产生；
2. 各阶段用于插入的公共边互不相同。

直观上，谱系记录了从三城市回路逐步扩展到 n 城市回路的历史。公共边互不相同对应“已经被删除的边不能再次被插入”。论文中的 Lemma 3.2 说明整数 MI 解的松弛变量可以解释为最终回路的边关联向量。

这里需要区分三个对象：

1. 三角形序列：插入历史；
2. 最终 Hamilton 回路：插入历史的边界结果；
3. 谱系特征向量：多面体空间中的 0-1 点。

不同插入历史是否可能产生同一回路、论文使用的投影是否保持优化等问题，都需要精确的双射或目标等价关系。论文引用作者早期工作处理这些关系，但 Lean 工程中的相关定理如后文所述并没有完整表达这些语义。

2. MI 整数规划

论文的 Pedigree Optimisation Problem 每层选择一个三角形：

$$\sum_{u \in \Delta_k} x_u = 1.$$

其他约束控制一条边可否在后续被选作插入边。若给定城市间距离 c_{ij}，定义插入成本：

$$C_{ijk} = c_{ik} + c_{jk} - c_{ij}.$$

这正是把城市 k 插入边 $\{i,j\}$ 导致的回路长度增量。因此总回路长度等于初始三角回路长度加所有阶段插入成本之和：

$$c_{12}+c_{13}+c_{23}+\sum_{i<j<k}(c_{ik}+c_{jk}-c_{ij})x_{ijk}.$$

如果整数可行解与 Hamilton 回路之间确实一一对应，那么在线性目标下优化所有谱系特征向量的凸包，确实等价于求最短 Hamilton 回路。这一局部成本分解是论文中最直观、也最容易理解的部分。

3. M3P

记所有谱系特征向量为 P_n，其凸包为：

$$\mathrm{conv}(P_n).$$

M3P 的输入是有理点 X，问题是：

$$X \overset{?}{\in} \mathrm{conv}(P_n).$$

成员问题不同于只检查 MI 松弛约束。满足线性松弛 $P_{MI}(n)$ 的点未必能表示成谱系的凸组合。要判定凸包成员性，需要识别所有隐含的全局一致性约束。

论文试图用分层网络完成这一任务。若成功，这将是实质性结果，因为一个多项式时间的精确成员 oracle 加上适当的有理性、满维性和内部点条件，可能通过已知凸优化理论转化为优化算法。

四、论文的分层网络与流方法

1. 基本网络结构

网络每一层对应一个插入阶段，节点对应该层具有正权重的边或三角形选择，节点容量来自输入点 X 的相应坐标。若一个选择可以生成下一层选择，则两节点之间连边。

如果 X 已经是谱系凸组合，那么每个谱系可以视为从第一层到最后一层的一条路径，谱系权重就是该路径携带的流。于是：

1. 节点边际等于 X 的层坐标；
2. 路径必须符合生成关系；
3. 所有路径的权重和为 1。

这个方向很自然：一个已知凸组合可以诱导出网络流。困难是反方向：一个满足若干边际和容量约束的流，是否一定可以分解成合法谱系，并且不同层之间是否保持同一个全局混合分布。

2. FAT 问题与刚性弧

论文在每个阶段构造禁弧运输问题（Forbidden Arc Transportation，FAT）。来源节点有供给，目标节点有需求，一些配对因生成关系不成立而禁止。

若某条弧在所有可行运输方案中流量都相同，则称为刚性弧；流量恒为零的刚性弧称为 dummy arc。论文使用 Frozen Flow Finding 算法识别刚性部分，并把唯一确定的路径压缩成刚性谱系 R_k，其权重记为 μ_P。

这个设计试图把分布分成两部分：

1. 已经由局部约束唯一决定的刚性部分；
2. 仍需要多商品流协调的灵活部分。

论文定义：

$$z_{\max} = 1 - \sum_{P \in R_k} \mu_P.$$

这代表除去刚性谱系权重后，剩余应由多商品流解释的总质量。

3. MCF 问题

论文为网络中的某些弧指定商品。每种商品只能沿特定受限网络流动，并满足：

1. 非负性；
2. 弧容量；
3. 中间节点流守恒；
4. 商品流总和等于共享弧流；
5. 节点容量；
6. 源可用量；
7. 末层容量；
8. 总流最大化。

论文将最优值记为 z^*，并提出判定问题：

$$z^* \stackrel{?}{=} z_{\max}.$$

其思想是：若所有剩余质量都能在各受限网络中一致传递，则灵活部分也能被解释为合法谱系混合。

4. 论文声称的充要条件

论文的 Theorem 6.10 声称，在相应前提下：

$$X \in \mathrm{conv}(P_n) \iff \mathrm{MCF}(n-1) \text{ 存在达到 } z_{\max} \text{ 的可行解}.$$

该双向命题是整篇论文真正的枢纽。后续所有复杂度结论都依赖它，而不只是依赖其中某一方向。

五、充分性与必要性为什么必须同时成立

1. 两个方向的不同功能

设判定条件为 $C(X)$：“MCF 达到 $z_{\max}$”；成员性为 $M(X)$：“$X \in \mathrm{conv}(P_n)$”。

充分性是：

$$C(X) \Rightarrow M(X).$$

它说明 MCF 成功时可以安全回答“是”。它排除了假阳性：算法不会把非成员点错误认定为成员点。

必要性是：

$$M(X) \Rightarrow C(X).$$

它说明所有真正的成员点都能通过 MCF 测试。它排除了假阴性：算法不会把成员点错误认定为非成员点。

一个决定算法必须对所有输入都给出正确答案，所以必须拥有双向等价。只有充分性时，MCF 至多是一个“成员证书生成器”或半判定条件，而不是完整成员算法。

2. 论文第 10.4 节的逻辑错误

论文明确说，P=NP 的机器验证链“只需要充分性方向”，因为从 MCF 达到最大流可推出成员性。这个说法不成立。

Tardos 算法可以求解给定线性规划并判断其最优值是否等于 $z_{\max}$。但如果只知道：

$$z^* = z_{\max} \Rightarrow X \in \mathrm{conv}(P_n),$$

那么当 $z^* < z_{\max}$ 时不能推出：

$$X \notin \mathrm{conv}(P_n).$$

因此不能把“是否达到 $z_{\max}$”当作 M3P 的正确判定程序。要从 MCF 的可解性推出 M3P∈P，必要性不可省略。

3. 论文中的必要性状态

论文第 6.9 节给出必要性证明的文字版本：假设 X 已经是谱系凸组合，以各谱系权重沿相应路径分配流，从而构造 MCF 解。这个思路在高层上合理，但其正确性依赖：

1. 每个成员凸组合中的活跃谱系都在递归删点后的受限网络中保留；
2. 刚性谱系的提取没有破坏质量守恒；
3. 商品划分覆盖所有非刚性质量且不重复；
4. 所有节点、弧、源和末层容量同时满足；
5. 网络是由输入 X 以确定、有效的方式构造；
6. 每一层的递归不依赖未知的凸组合。

论文把若干关键支持结论放在附录并多次引用作者的书。即使接受文字证明，论文自己也承认必要性尚未完成 Lean 形式化。公开仓库的核心文件更明确写道：

```
axiom necessity_of_mcf ...
```

并注明备份文件仍有 16 个 `sorry` 。因此“完整充要条件已经机器验证”与源码事实不一致。

六、Lean 4 工程的核心规格审计

1. 机器验证验证的首先是“类型”

Lean 内核验证的是：某个项是否具有某个类型。若类型准确表达数学定理，且证明只依赖可信基础和明确外部定理，那么机器验证具有很强意义。

但如果把待证结论直接写成公理：

```
axiom target : Prop
```

或者把复杂数学对象写成占位类型：

```
KW1 Polytope := Unit
```

Lean 仍然可以无 `sorry` 地完成后续推导。此时机器验证保证的是“在这些假设下推理合法”，不保证假设真实，也不保证占位对象对应原数学对象。

因此审计不能只搜索 `sorry` ，还必须检查：

1. 定义；
2. 定理结论；
3. 结构字段；
4. 公理清单；
5. 公理的强度；

6. 论文描述与源码类型是否一致。

2. `MCFFeasible` 直接包含凸组合见证

仓库文件 `N_LayeredNetworkTypes.lean` 中：

```
structure MCFFeasible ... where
  ...
  conv_wit : ConvexWitness n k X
```

这意味着一个 `MCFFeasible n k net X` 的实例，不只是提供流变量及其约束，还要求调用者预先提供：

$$X/k \in \mathrm{conv}(P_k)$$

的凸组合见证。

论文中的递归定理确实可能把“前一层属于凸包”作为归纳前提，但这里存在两个严重问题。

第一，若 MCF 可行性作为算法判定对象，其输入规格不能包含一个通常难以获得的凸组合见证，除非论文证明该见证可由此前算法步骤多项式构造并作为显式状态维护。仅把见证塞进结构字段，不等于给出构造算法。

第二，论文把 MCF 描述为一个线性规划，其变量应主要是流量。 `ConvexWitness` 是组合对象及正权重表示，不是论文 Problem 6.2 所列的线性规划变量。Lean 中的 `MCFFeasible` 因而比论文中的“MCF 约束可行”强得多。

这会产生规格偷渡：本来要由流推出的凸包信息，已经部分进入流可行性的定义。

3. 所谓充分性证明没有完成论文描述的构造

论文中的 Theorem 6.8 说：

1. 对每种商品 s，由流构造归一化点 Y^s/v^s；
2. 证明它属于 $\mathrm{conv}(P_k)$；
3. 将每个谱系沿商品末端扩展到 P_{k+1}；
4. 给这些扩展谱系权重 $v^s\gamma_P$；
5. 加入刚性谱系及权重 μ_P；
6. 证明总权重为 1；
7. 证明加权特征向量之和恰为 $X/(k+1)$。

但仓库 `N_Sufficiency.lean` 中的实际结论是：

```
theorem sufficiency ... :
  ∃ wit : ConvexWitness n (k + NUM1) X, True := by
```

证明的关键步骤是：

```
obtain ⟨s, hs⟩ := List.exists_mem_of_ne_nil _ hne
obtain ⟨wit, _⟩ :=
  Y_s_in_conv net mcf ... s hs mcf.conv_wit
exact ⟨wit, trivial⟩
```

它只选择一个商品，调用 `Y_s_in_conv`，并把得到的见证返回。源码注释自己列出了三个原本需要完成的数学任务：

1. SF1：扩展每个谱系；
2. SF2：证明组合权重和为 1；
3. SF3：证明组合恰好等于 $X/(k+1)$。

然而实际证明没有实现“对全部商品求和并加入刚性谱系”的构造。这一点不是代码风格问题，而是定理内容是否对应论文证明的问题。

4. `Y_s_in_conv` 使用的是输入见证，不是由流构造的 Y^s

`N_YisinConv.lean` 中：

```
theorem Y_s_in_conv ...
    (hwit : ConvexWitness n k X) :
    ∃ wit : ConvexWitness n (k + NUM1) X, True := by
```

它使用 `hwit` 作为 X/k 的凸组合，并借助 `src_val` 等条件调用 Packability Corollary。这里证明的是一种特殊扩展：

1. 已经知道整个 X/k 的凸组合；
2. 某个末层坐标等于 1；
3. 因此所有活跃谱系都能用同一个插入选择扩展。

但论文一般情形中，各商品只承载总质量的一部分 v^s，不同商品对应不同末端选择。通常不会有整个 X 在每个商品的 `arc_head` 上都满足坐标等于 1。真正需要归一化的是商品诱导的子点 Y^s/v^s，而不是原始 X。

Lean 结构却要求：

```
src_val : ∀ s ∈ commodities, X s.arc_head = NUM1
```

这比论文中的一般 MCF 情况强得不自然。如果存在多个具有不同 `arc_head` 的商品，要求原始层向量在每个相应坐标上都等于 1，与每层坐标和为 1 通常不能同时成立。除非所有商品头相同或结构退化，否则该规格可能不可满足。

因此形式化并没有忠实表达论文定义的商品子流归一化逻辑。

5. “无 sorry”不能掩盖公理和弱规格

论文反复强调主链“zero sorrys”。但一个可靠的形式化审计至少要统计：

1. `sorry`；
2. `axiom`；
3. `opaque` 或占位定义；
4. 定理是否只返回 `True`；
5. 结论是否比自然语言弱；
6. 假设是否已经包含结论。

当前工程在核心目录中存在大量公理。即使这些公理在 Lean 中不是 `sorry`，它们同样是未由内核从更基础定义证明的外部假设。更重要的是，其中不少不是像“Cook-Levin 定理”那样对标准形式对象的精确陈述，而是对不透明命题的直接蕴含。

所以“没有 sorry”不是“从标准定义开始完整验证”的同义词。

七、P、NP 与算法复杂度并未被形式化

1. 标准形式化应包含什么

要在 Lean 中真正表达 P=NP，至少需要：

1. 输入字符串或有限编码；
2. 判定问题或语言；
3. 确定性和非确定性计算模型；
4. 算法接受语义；
5. 运行步数；
6. 输入长度；
7. 多项式时间界；
8. 复杂度类 P 与 NP；
9. 多项式时间 many-one 归约；
10. NP 完全性。

形式化可以采用图灵机、RAM、递归函数配合成本模型，或复用成熟复杂度库。具体模型可以不同，但必须有明确语义并证明模型间必要等价。

2. 仓库中的真实定义

`N_PEqualsNP.lean` 中并没有定义这些概念，而是写：

```
axiom STSP_in_P : Prop
axiom P_equals_NP : Prop
axiom SAT_in_P : Prop
```

也就是说：

1. `STSP_in_P` 只是一个没有内部结构的命题符号；
2. `SAT_in_P` 只是另一个命题符号；
3. `P_equals_NP` 也是一个命题符号。

最终定理：

```
theorem p_equals_np ... : P_equals_NP
```

证明的并不是两个已定义复杂度类相等，而是推出一个名为 `P_equals_NP` 的抽象命题。

名称不会赋予命题数学语义。把一个命题命名为 `P_equals_NP` ，与定义真实的 P 和 NP 并证明其相等，是完全不同的事情。

3. Cook 与 Karp 被包装成不透明蕴含

源码写：

```
axiom cook_np_completeness : SAT_in_P → P_equals_NP
axiom karp_stsp_np_complete : STSP_in_P → SAT_in_P
```

如果 `SAT_in_P` 、 `STSP_in_P` 、 `P_equals_NP` 都没有语义定义，那么 Lean 只能验证两次函数应用：

$$STSP\_in\_P \Rightarrow SAT\_in\_P \Rightarrow P\_equals\_NP.$$

它没有检查：

1. STSP 实例编码是否正确；
2. 判定版和优化版转换；
3. Karp 归约是否多项式；
4. SAT 是否为真正的布尔可满足性问题；

5. P 与 NP 的定义。

引用 Cook 和 Karp 的论文不能弥补形式规格缺失。外部定理公理必须精确地作用于真实定义，才能把其可信度传递给最终结论。

八、 `A_n := Unit` 与多面体优化链失真

1. 论文中的 A_n

论文引入 A_n 作为谱系特征向量删去每层一个坐标后的投影集合。其凸包应位于维数：

$$\alpha_n = \tau_n - (n-3)$$

的有理向量空间中。论文需要证明：

1. $\mathrm{conv}(A_n)$ 满维；
2. 有理性或面复杂度有多项式界；
3. 已知一个内部点；
4. 从 $\mathrm{conv}(P_n)$ 的成员协议得到 $\mathrm{conv}(A_n)$ 的成员协议；
5. 由成员得到分离，再由分离得到优化。

这些性质用于调用 Maurras 和 Grötschel-Lovász-Schrijver 的结果。

2. Lean 中的 A_n

核心源码写：

```
KW1 An (n : ℕ) : Type := Unit
```

`Unit` 只有一个元素。它不是 $\mathbb{Q}^{\alpha_n}$ 中的谱系投影点集，也没有论文所需的坐标、凸包、面、内部、线性目标或编码长度结构。

随后源码把性质写成：

```
axiom FullDimensional (P : Type) : Prop
axiom RationalityGuaranteed (P : Type) : Prop
axiom HasInteriorPoint (P : Type) : Prop
```

并进一步公理化：

```
axiom convAn_full_dimensional_ax ...
axiom convAn_rationality_guaranteed_ax ...
axiom convAn_has_interior_point_ax ...
```

这些命题没有连接到实向量空间中的凸几何定义。因而所谓“满维”“内部点”“有理性”只是标签，不是被 Lean 检查的几何性质。

3. 成员到分离再到优化

源码中的：

```
axiom QuickProtocol (P : Type) : Prop
axiom PolynomialSeparationOracle (P : Type) : Prop
axiom PolynomialOptimisation (P : Type) : Prop
```

同样只是抽象命题。没有算法、oracle 接口、正确性条件、运行时间函数或位复杂度。

Maurras 和 GLS 被写成这些抽象命题间的公理化蕴含。即使外部数学定理本身正确，当前 Lean 文件并没有证明其前提对应真实多面体，也没有表达其复杂度结论的定量内容。

因此第 4—5 步不是“机器验证引用了经典定理”，而是“机器接受了作者声明的抽象蕴含”。

九、M3P∈P 与 $O(n^{14})$ 复杂度声明

1. 论文的复杂度路线

论文估计：

1. 每层受限网络节点数和弧数是 n 的多项式；
2. 每个 link 的最大流可多项式求解；
3. link 数量为多项式；
4. FAT 和刚性识别为多项式；
5. 刚性谱系数量由维数界控制；
6. MCF 约束矩阵元素属于 $\{0, \pm 1\}$；
7. 使用 Tardos 算法得到强多项式求解；
8. 总算术步数为 $O(n^{14})$。

这里至少要分别证明：

1. 数据结构规模界；
2. 构造这些数据结构的算法界；

3. LP 变量和约束数量界；
4. Tardos 定理适用条件；
5. 算术操作与位复杂度；
6. 判定结果与 M3P 等价。

只证明其中某个集合的基数是多项式，不能自动得到整个算法属于 P。

2. Lean 中 `compexity` 的实际结论

源码中名为 `compexity` 的定理实际证明：

```
net.rigid.length ≤ tau n - k + NUM1
```

名为 `M3P_in_P` 的定理实际也是：

```
net.rigid.length ≤ tau n - n + NUM1
```

这只是刚性谱系列表的长度界。它没有声明或证明：

1. 一个函数接收编码后的 X；
2. 函数终止；
3. 函数输出成员或非成员；
4. 输出正确；
5. 运行时间小于某个多项式；
6. 算法在所有输入上适用。

因此论文把 Lean 中的基数定理描述成“M3P∈P 的机器证书”，是不准确的。

3. Tardos 公理包含待证结论

源码直接写：

```
axiom tardos_strongly_polynomial ... :
  QuickProtocol (LayeredPoint n)
```

但 Tardos 1986 的定理不是“任意 `LayeredPoint n` 的成员问题都有 quick protocol”。它针对特定组合线性规划的求解复杂度。要得到这里的结论，必须先完成从 M3P 到具体 LP 的正确归约。

当前公理省略了 LP：

1. 没有矩阵 A、右端 b、目标 c；

2. 没有矩阵元素界；
3. 没有规模参数；
4. 没有算法输出；
5. 没有成员性双向等价。

所以它不是对 Tardos 定理的精确形式化，而是把“由 Tardos 推出 quick protocol”整体作为公理。这个公理已经包含论文争议最大的步骤之一。

4. 强多项式与普通多项式

论文还声称运行时间独立于 X 数值大小，因此是强多项式。即使 LP 核心可用 Tardos 算法，外围构造仍处理有理容量、刚性流权重和精确相等比较。完整论证需要说明：

1. 算法执行的算术操作数只依赖组合维数；
2. 中间数的编码长度保持多项式；
3. 输入读取本身和输出编码受到控制；
4. 比较 $z^* = z_{\max}$ 可精确完成；
5. 网络递归所需的数值不会引入额外位复杂度。

论文给出的是高层计数式，并没有在 Lean 中形式化这些内容。因此即使暂不考虑成员充要性，$O(n^{14})$ 的“机器验证”也不成立。

十、从谱系优化到 STSP 的形式化缺口

1. 数学上需要证明的内容

从谱系多面体线性优化到 STSP，至少需要：

1. 每个谱系对应合法 Hamilton 回路；
2. 每个 Hamilton 回路至少对应一个谱系，或证明优化范围等价；
3. MI 目标加初始成本恰等于该回路成本；
4. 投影 A_n 不丢失计算目标所需信息；
5. 从优化 oracle 的输出恢复实际回路；
6. 对有理距离输入，算法编码和运行时间为多项式。

论文通过早期 MI 形式研究和 Lemma 3.2 解释这一关系。作为非形式化数学路线，这部分有一定合理基础。但“Lean 已经验证”需要真实表达上述对象和映射。

2. Lean 中的 `lemma_oneone`

源码中的定理为：

```
theorem lemma_oneone ...
    (hX : ∀ t, X t = NUM1 ∨ X t = NUM2) :
    ∃ tour : Finset (ℕ × ℕ), True
```

结论只说存在某个有限边集，并附带 `True` 。它没有证明该边集：

1. 每个顶点度数为 2；
2. 连通；
3. 覆盖所有城市；
4. 是由输入 X 对应的回路；
5. 与松弛变量一致；
6. 成本等于 MI 目标。

任何有限边集都满足 `True` ，甚至空集也可以作为见证。因此这不是论文所需的一一对应定理。

3. `mi_objective_solves_stsp` 依赖公理

源码最后写：

```
axiom mi_objective_solves_stsp_ax ...
    (h_opt : PolynomialOptimisation (An n)) :
    STSP_in_P
```

随后所谓定理只是调用这个公理。它没有利用一个已证明的一一对应关系，也没有实现或验证成本等价。

因此“步骤 6 已证明”的说法与源码不符。该步骤实质上仍是外部假设。

十一、论文关于“六个公理”的陈述是否准确

论文列出的公理包括 Tardos、Maurras、GLS、Cook、Karp 和 Rao，试图传达：只有六个广泛接受的外部结果作为公理，其他内容都已内部证明。

但核心源码中还存在：

1. `QuickProtocol` ；
2. `PolynomialSeparationOracle` ；

3. `PolynomialOptimisation`；
4. `FullDimensional`；
5. `RationalityGuaranteed`；
6. `HasInteriorPoint`；
7. `STSP_in_P`；
8. `P_equals_NP`；
9. `SAT_in_P`；
10. `convAn_full_dimensional_ax`；
11. `convAn_rationality_guaranteed_ax`；
12. `convAn_has_interior_point_ax`；
13. `mi_objective_solves_stsp_ax`；
14. `membership_An_of_Pn`；
15. `necessity_of_mcf`；
16. 若干谱系构造支持公理。

需要说明：把一个“谓词符号”声明为公理不必然导致不一致，例如 `axiom QuickProtocol (P : Type) : Prop` 只是为每个类型假设一个命题成立。但它会使后续相关结论失去证明价值，因为所需性质从一开始就被无条件授予。

更严重的是，论文声称某些条件由 Lean 证明，源码却以专门公理直接提供。例如 `convAn_full_dimensional_ax` 直接给出不透明 `FullDimensional (An n)`，而注释中提到的某个超平面系数定理并没有通过已证明的等价桥接到这个谓词。

因此公理库存不能只按引用文献名称计数，必须按 Lean 环境中的实际常量及其类型审计。按这一标准，“仅六个外部公理”明显不准确。

十二、论文中可保留的研究内容

否定 P=NP 结论并不意味着论文每个局部想法都没有价值。应当区分以下层次。

1. 逐阶段插入表示

用三角形记录城市插入已有边的过程，是一种清晰的 Hamilton 回路编码。插入成本：

$$c_{ik} + c_{jk} - c_{ij}$$

也具有直接意义。这种表示适合研究递归结构、生成关系和历史约束。

2. 谱系多面体

由插入历史特征向量取凸包，可以形成区别于经典边空间 TSP 多面体的扩展或替代表示。其：

1. 维数；
2. 邻接关系；
3. 面结构；
4. 投影；
5. 与 Hamilton 回路多面体的联系；
6. 松弛强度；

都可能是合理的组合多面体研究问题。

3. 刚性流思想

在一组具有固定边际的运输方案中识别所有可行解都保持不变的弧，并递归压缩这些确定部分，是有意义的算法思想。它可能用于：

1. 分析某些边际分布的唯一分解；
2. 生成成员性的充分证书；
3. 发现非成员性的局部障碍；
4. 设计启发式分解算法。

4. 成员性证书

如果某个 MCF 解确实能够被严格转换为谱系凸组合，那么它至少提供一种成员证书。即使条件不是必要的，一个多项式可验证的充分条件仍有应用价值，例如：

1. 快速确认一部分成员点；
2. 构造显式凸组合；
3. 辅助分支算法；
4. 研究特定子类或特殊实例。

但这类结果应准确表述为“充分条件”或“证书系统”，不能称为完整成员算法。

5. 形式化尝试的正面意义

作者尝试把复杂的组合优化论证形式化，这一方向值得肯定。形式化过程实际上帮助暴露了问题：

1. 必要性尚有多个 `sorry`；
2. 充分性真正需要组装所有商品和刚性质量；
3. 复杂度概念需要真实计算模型；

4. 多面体投影不能用 `Unit` 替代；
5. 自然语言中的“显然”需要明确接口。

因此该工程可作为一个尚未完成的形式化研究原型，而不能作为 P=NP 证书。

十三、论文没有证明什么

根据目前论文和仓库，以下结论没有被建立：

1. 没有证明标准意义的 P=NP

原因不是简单的“学界还没接受”，而是证明链的核心逻辑和形式规格均不完整。

2. 没有机器验证 P 与 NP 两个复杂度类相等

工程没有定义这两个复杂度类，最终目标只是公理化命题 `P_equals_NP` 。

3. 没有机器验证 M3P∈P

名为 `M3P_in_P` 的定理只给出列表长度界；真正的 quick protocol 被 Tardos 名义下的公理直接提供。

4. 没有机器验证 MCF 的完整充要性

必要性在核心文件中是公理；备份证明尚有 `sorry` 。充分性实现也没有完成论文描述的多商品组合构造。

5. 没有机器验证谱系优化等价于 STSP

一一对应定理被弱化为“存在有限边集且 True”，最终 STSP∈P 由公理给出。

6. 没有机器验证 $O(n^{14})$ 算法

工程没有一个带成本语义的可执行成员算法，也没有证明运行时间函数。

7. 没有证明密码学立即失效

论文第 11 节关于密码学的讨论建立在 P=NP 已成立这一前提上。由于主结论未建立，这些“立即后果”没有现实依据。即使未来证明 P=NP，也仍需考虑多项式次数、常数、平均复杂度、搜索到判定转换和具体安全参数，不能仅凭类包含关系断言所有现实系统立即可破。

十四、如果要把该工作修复为可信研究，需要完成什么

1. 首先降低主张范围

在当前状态下，合理标题应类似：

> A Layered Multicommodity-Flow Sufficient Condition for Membership in the Pedigree Polytope

或：

> Towards a Formalisation of Pedigree Polytope Membership in Lean 4

在完整充要性、算法和复杂度链建立前，不应声称 P=NP。

2. 重写 MCF 形式规格

`MCFFeasible` 应只包含论文 Problem 6.2 中的：

1. 流变量；
2. 商品集合；
3. 容量；
4. 流守恒；
5. 商品限制；
6. 目标值条件。

不应直接包含 `ConvexWitness`，除非它明确作为递归算法已构造状态，而且定理区分：

1. 输入中已有的前层见证；
2. 当前层由流新构造的见证。

尤其应定义真正的商品子点 Y^s，证明：

$$\frac{1}{v^s}Y^s \in \mathrm{conv}(P_k),$$

而不是对原始 X 使用每个商品头坐标等于 1 的条件。

3. 完成真正的充分性证明

形式化必须构造一个总索引类型，例如：

$$\{(s,r) : s \in S_k,\ r \in I_s\} \sqcup R_k,$$

定义：

1. 商品部分谱系扩展；

2. 商品部分权重 $v^s\gamma_{s,r}$；
3. 刚性谱系权重 μ_P；

并证明：

$$\sum_{s,r} v^s\gamma_{s,r} + \sum_{P\in R_k} \mu_P = 1,$$

以及对每个坐标 t：

$$\sum_{s,r} v^s\gamma_{s,r}\mathbf{1}[t \in P'_{s,r}] + \sum_{P\in R_k} \mu_P\mathbf{1}[t \in P] = X_t.$$

这正是源码注释中的 SF1—SF3，不能以选取单个商品代替。

4. 完成必要性且消除循环

必要性需要从任意 `ConvexWitness` 构造纯流解，并证明流解满足全部受限网络约束。尤其要证明网络构造只依赖输入 X 和此前有效计算结果，而不依赖正在证明存在的未知凸组合。

若某一步需要选择特定凸分解 λ，还必须说明算法如何获得它，或证明最终构造与该选择无关。

5. 形式化真实复杂度概念

可以不从零开始构造全部理论，但至少应：

1. 选择明确的输入编码；
2. 定义 M3P 判定函数；
3. 证明函数正确；
4. 定义成本模型；
5. 证明数据结构规模和运行步数界；
6. 精确实例化 Tardos 定理。

“矩阵元素为 $0,\pm1$”只是适用条件之一，不等于自动获得整个递归算法的强多项式性。

6. 形式化真实多面体 A_n

不能再使用 `Unit` 。应把 A_n 定义成有限维有理或实向量空间中的有限点集或其凸包，并定义投影线性映射。然后证明：

1. 投影与逆恢复关系；
2. 满维性；
3. 内部点；
4. 有理面复杂度；

5. 成员 oracle 转移；
6. 目标函数转移。

7. 形式化 STSP 与归约

应定义：

1. 有限城市集；
2. 对称有理成本矩阵；
3. Hamilton 回路；
4. 回路成本；
5. STSP 判定问题；
6. 谱系到回路映射；
7. 回路到谱系映射或优化等价；
8. 多项式时间恢复。

只有完成这些内容， `STSP_in_P` 才能是定理，而不是命题标签。

8. 最后再连接标准 P 与 NP

最好复用已有复杂度理论形式化库，或明确实现语言、验证器和归约。然后引用已形式化的 Cook-Levin 和 TSP NP 完全性结果。若只能把经典结果作为公理，也必须使公理作用于真实定义，而不是无语义的抽象 `Prop` 。

十五、总体评价

1. 对论文数学主线的评价

论文最有价值的部分是把谱系多面体成员问题组织成分层网络、刚性运输和多商品流框架。它试图解决一个真实困难：如何在不知道凸分解的情况下，从局部层边际恢复全局谱系混合。

但论文从该框架跳到 P=NP 的关键台阶没有站稳。主要问题不是某个边角计算，而是：

1. 成员判定的必要性不可省略；
2. 流可行性规格与凸组合见证发生循环；
3. 充分性构造没有按宣称完成；
4. 复杂度定理没有建立算法语义；
5. 成员到优化的前提没有被真实建模；
6. STSP 等价关系没有被实际形式化。

这些问题直接影响最终结论，而非只影响表达严谨性。

2. 对机器验证声明的评价

论文把“Lean 编译通过”“没有 sorry”和“数学结论已验证”视为近乎等价，这是对证明助理保证范围的误解。

Lean 保证：

> 在当前环境的定义、公理和假设下，给定证明项符合给定类型。

Lean 不自动保证：

1. 类型忠实表达自然语言命题；
2. 公理来自被正确陈述的外部定理；
3. 名称为 `P_equals_NP` 的命题就是标准 P=NP；
4. 注释中的证明已经体现在代码里；
5. 一个名为 `M3P_in_P` 的定理真的表达 M3P∈P；
6. 占位类型 `Unit` 是某个高维多面体。

当前工程恰好在这些方面存在系统性问题。因此它不能作为论文主结论的独立证书。

3. 可信度分层

可以把论文内容分为四层：

第一层：基本建模直觉。 逐阶段插入、三角形表示、插入成本分解，具有清晰直觉和既有研究背景。

第二层：局部组合与流性质。 某些 FAT 可行性、刚性弧、谱系邻接和成员充分条件可能成立，但需要逐一定理审查。

第三层：完整 M3P 强多项式算法。 目前没有被论文和 Lean 工程充分建立。

第四层：STSP∈P 与 P=NP。 依赖第三层及多个未形式化桥接，当前不成立。

4. 最终结论

本报告的最终判断是：

> 该论文不能被视为 P=NP 的证明，也不能被视为 P=NP 的 Lean 4 机器验证。公开形式化工程通过公理化、占位定义、把目标信息放入结构假设以及弱化定理结论，使 Lean 能够接受一个与论文自然语言主张显著不同的证明链。

更具体地说：

1. 论文明确宣称的 MCF 充要性没有完整机器验证；
2. 声称“只需充分性即可推出 M3P∈P”是逻辑错误；
3. Lean 的充分性没有实现论文中的全商品凸组合构造；
4. `MCFFeasible` 预先携带凸组合见证，削弱了证明内容；
5. `P_equals_NP` 只是公理化命题，不是已定义复杂度类的等式；
6. A_n 被定义成 `Unit`，凸几何链没有真实形式化；
7. `M3P_in_P` 实际只证明刚性集合基数界；
8. STSP∈P 及 MI 优化等价由公理而非证明获得；
9. “仅六个外部公理”和“主链完整验证”的描述与仓库事实不符。

因此，最准确、最直观的一句话总结是：

> 论文提出了一个值得继续研究的谱系多面体流框架，但其 P=NP 结论既未由数学论证完整推出，也未被 Lean 4 按标准复杂度理论语义验证。

附录 A：关键证据对照表

论文声称	仓库实际内容	审查结论

P=NP 已由 Lean 验证	`axiom P_equals_NP : Prop`，最终定理返回该抽象命题	未定义标准 P、NP
M3P∈P 已验证	`M3P_in_P` 结论是刚性列表长度上界	名称与定理内容不符
MCF 充分性已构造凸组合	取一个商品并复用 `mcf.conv_wit`	没有组装全部商品和刚性部分
MCF 可行性是纯流条件	`MCFFeasible` 含 `conv_wit : ConvexWitness`	把凸包见证放入假设
MCF 充要性已建立	必要性是 `axiom necessity_of_mcf`	机器验证未完成
A_n 的满维、内部点等已验证	`An n := Unit`，性质是不透明公理	没有真实凸几何语义
MI 形式解决 STSP 已证明	`mi_objective_solves_stsp_ax` 直接给出 STSP∈P	关键桥接被公理化
一一对应已证明	结论为 `∃ tour, True`	不表达 Hamilton 回路性质
Tardos 推出 quick protocol	直接公理 `tardos_strongly_polynomial : QuickProtocol ...`	未精确实例化 Tardos 定理
仅使用六个外部公理	核心目录还有多项额外公理和占位谓词	公理库存陈述不准确

附录 B：核心源码位置

以下位置基于研究时获取的公开仓库版本：

1. `MembershipProject/Core/N_PEqualsNP.lean`
    1. 第 57—85 行：复杂度与凸几何性质被定义为公理谓词。
    2. 第 102—111 行：`STSP_in_P`、`P_equals_NP`、`SAT_in_P` 为公理命题。
    3. 第 145—146 行：`An n` 被定义为 `Unit`。
    4. 第 182—224 行：Tardos、Maurras、GLS、Cook、Karp 被包装成抽象公理。
    5. 第 236—258 行：`conv(A_n)` 性质和 STSP 结论的额外公理。
    6. 第 318—320 行：投影成员协议转移是公理。
    7. 第 335—338 行：所谓一一对应只产生有限边集和 `True`。
    8. 第 364—382 行：最终 `p_equals_np` 主要串联上述公理。

1. `MembershipProject/Core/N_LayeredNetworkTypes.lean`
    1. `MCFFeasible` 中包含 `conv_wit : ConvexWitness n k X` 。
    2. `src_val` 要求每种商品对应的原始 X 坐标等于 1。
    3. `flow_is_zMax` 作为结构字段直接提供。
1. `MembershipProject/Core/N_Sufficiency.lean`
    1. 第 39—55 行注释承认真正构造需要 SF1—SF3。
    2. 第 83—129 行实际证明只选一个商品并返回已有链条产生的 witness。
1. `MembershipProject/Core/N_MembershipCharacterisation.lean`
    1. 第 12—19 行明确必要性是公理且备份证明有 16 个 `sorry` 。
    2. 第 55—62 行声明 `necessity_of_mcf` 公理。
    3. 第 86—108 行的充要定理依赖该公理。
1. `MembershipProject/Core/N_complexity.lean`
    1. 第 266—270 行的 `compexity` 只证明刚性列表长度界。
    2. 第 285—289 行的 `M3P_in_P` 同样只证明长度界。

附录 C：阅读本论文时最容易混淆的三个概念

1. “Lean 接受”与“数学主张被忠实验证”

Lean 接受证明项，只保证形式推理相对于当前环境正确。定义与公理的真实性仍需要人类审计。

2. “充分条件”与“判定算法”

充分条件可以安全确认部分“是”实例；判定算法必须对所有“是”和“否”实例都正确，因此通常需要充要条件。

3. “LP 可强多项式求解”与“原问题可强多项式求解”

只有在原问题与 LP 之间存在可计算、双向正确、规模多项式受控的归约时，LP 的复杂度结论才能传递给原问题。