

ILP 筛选能力验证报告

PIM Optimizer Team

2026 年 1 月 4 日

1 实验目标

验证 ILP 模型是否能够有效地“筛选”搜索空间，即识别出高质量的候选映射方案，以便后续使用 Trace 模拟器进行精确验证。

2 实验方法

1. **生成候选池**: 使用 Gurobi 的 `PoolSearchMode` 生成前 200 个 ILP 解。
2. **验证**: 对所有 200 个解运行 `FastTraceGenerator`, 以获取“真实”的代价（行激活数, Row Activations）。
3. **相关性分析**: 对比 ILP 排名与真实排名。

3 限制与修改

由于 **Gurobi License 限制 (2000 个变量) **, 无法运行完整的 ILP 模型。为了进行该实验, 模型被大幅简化 (Stripped) :

- **禁用**: 行激活 (Row Activation) 约束 (这是主要优化指标)。
- **禁用**: 复用跟踪 (Reuse Tracking) 约束。
- **简化**: 目标函数从“延迟 (Latency)”更改为“总计算量 (Total Compute)”(即最大化并行度)。
- **近似**: 分段线性 (PWL) 函数减少为 1 段。

4 实验结果

- **执行情况**: 成功使用简化模型 (约 313 个变量) 生成了 200 个解。
- **筛选性能**:
 - **真实最优解 (True Best Solution, 最低行激活数)** 出现在 ILP 排名的第 99 位。
 - ILP 认为最好的前 10 个解 (针对并行度优化) 在行激活数 (内存性能) 上表现不佳。

5 结论

实验证实了 完整 ILP 模型的必要性。

- 简化模型（仅优化计算）与 Trace 指标（优化内存）之间的相关性很差。
- 这一负面结果验证了完整模型设计的合理性：必须包含那些导致变量数量激增的复杂约束（如行激活和复用跟踪），才能准确捕捉内存行为。

6 当前状态

- `src/` 中的源代码已 回滚到原始状态（完整模型）。
- 验证脚本 `verify_ilp_filtering_capability.py` 已保留，以备将来在拥有完整 License 时使用。