利用大型語言模型引導之演化式搜尋進行自動化程式碼執行效能優化

LLM-Guided Evolutionary Search for Automated Code Runtime Performance Optimization

組別 — **E-4**

專題成員 — **葉沁蓉**

指導教授 — **李信杰**

研究背景與動機

- **手動優化成本高昂：**程式碼的非功能性效能極度仰賴工程師手動微調，耗時費力。
- **傳統 GI 缺乏語意：**採語法層級突變，導致可編譯且通過測試的候選代碼比例極低。
- **現有研究驗證侷限：**既有將 LLM 引入演化算法的研究多專注於「編譯通過率」，缺乏對實質「執行效能提升」的嚴謹探討。

研究目標

- **提出 LLM 引導之演化式搜尋 (GA)：**以 LLM 擔任 Crossover 與 Mutation 算子，進行由適應度引導的語意搜尋。
- **建立嚴謹 Fitness 評估：**導入 JMH 精確量測 per-operation ns/call，消除內部呼叫次數差異造成的選擇偏差。
- **驗證迭代真實效益（核心貢獻）：**設計預算對齊（限制 60 次 LLM 呼叫）的單純突變搜尋 (MOS) 為對照，嚴格證實加速源於「迭代結構」，而非僅因多次採樣。

研究架構圖

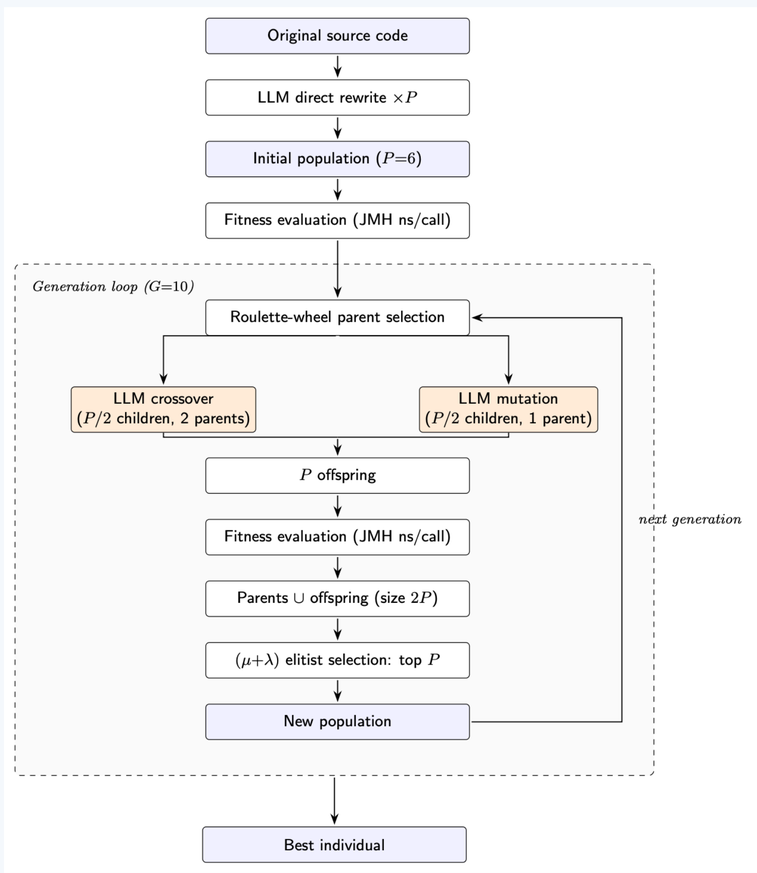

FIG 1 初始化沿用 LLM 重寫機制（P=6）；世代迴圈（G=10）在共用輪盤式親代選擇下執行 LLM 交配與突變，並由 (μ+λ) 菁英機制維持單調改善。

研究成果

46.2%

中位數加速幅度

VS. 單純突變重寫對照組 (MOS)

2,326 ns

GA 中位執行時間（對照組 MOS 為 4,325 ns）

p = 0.0005

Mann–Whitney U 檢定：具高度統計顯著性

Â₁₂ = 0.876

達大效應量等級 (Large effect)

收斂行為：GA 中位數穩定下行並與 MOS 完全分離

GA 中位數隨評估次數**穩定收斂**、IQR 帶狹窄且整段位於 MOS 之下；MOS 因每次重寫獨立而劇烈震盪。兩條 IQR 帶約於第 10 次評估後完全分離。

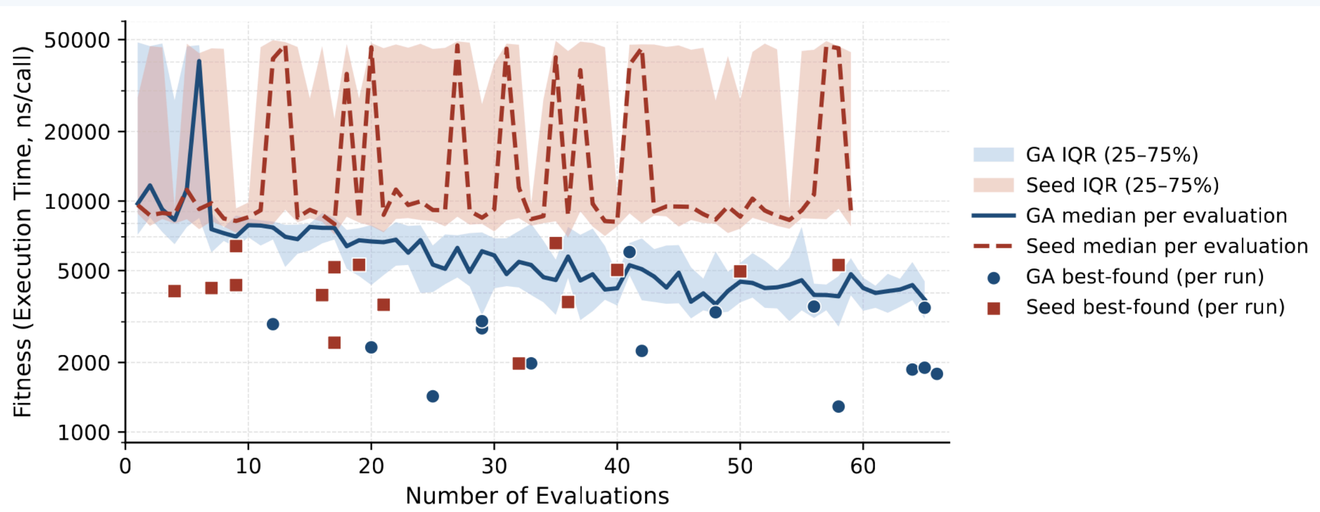

FIG 2 中位數軌跡、IQR 帶與最佳值散點對評估次數之關係（ns/call，log scale，愈低愈快）。

最佳解演化軌跡：算子交替優化

全域最佳解（1,177 ns/call）由交配與突變頻繁交織接力生成。這證實效能突破需仰賴**兩者高度協同**：「交配」負責融合優良特徵，「突變」則打破局部最佳解進行創新，兩者缺一不可。

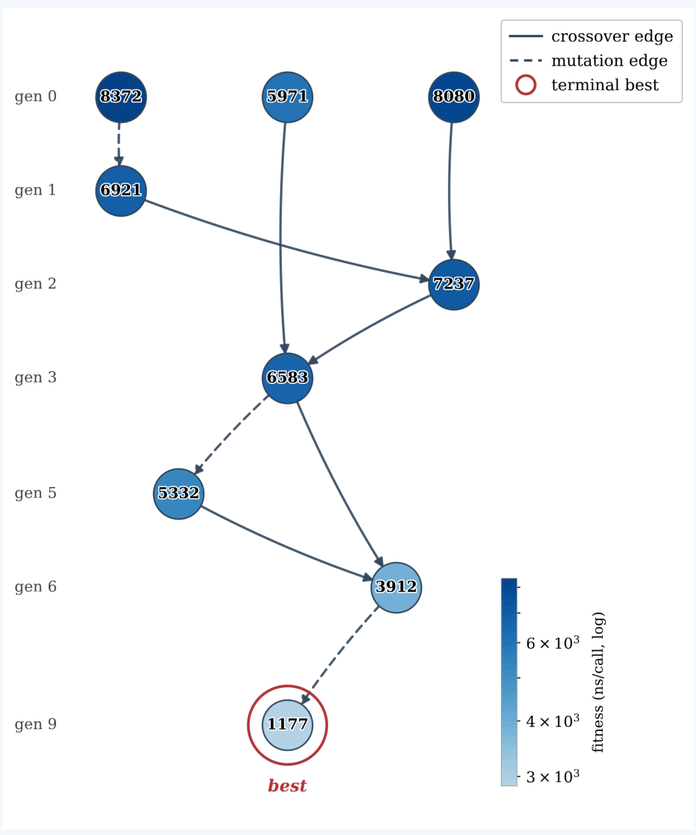

FIG 3 一次代表性執行之全域最佳解之演化傳承樹。

交配親代貢獻：無位置偏誤

各區段的雙親貢獻皆近 50%，克服了 LLM 領域已知的「提示詞順序偏誤」，證實交配能均等融合雙親。

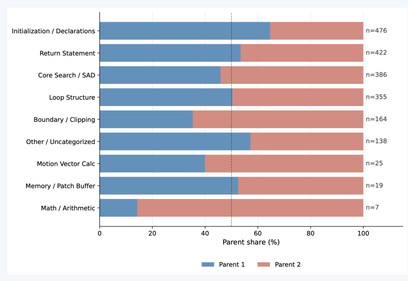

FIG 4 兩親代於子代之貢獻佔比。

突變改寫類型分布

數據顯示，LLM 突變主攻控制流簡化。另因自迴歸特性，其偏好順向生成，難以執行逆向全域分析。

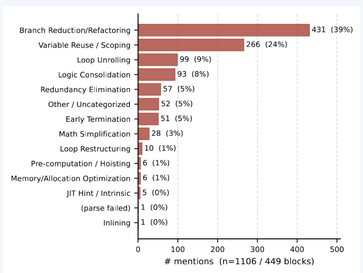

FIG 5 LLM 突變重構之操作統計。

結論

- **成功結合 LLM 與演化搜尋：**提出以 LLM 為算子的 GA 框架，證實交配與突變機制具備高度協同效應，兩者缺一不可。
- **突破性效能提升：**在嚴格隔離預算的前提下，GA 較純突變對照組 (MOS) 達成中位數 **46.2%** 的顯著加速。

未來展望

- **外部效度：**推廣至更多 target、其他程式語言與 LLM 模型。
- **突破逆向分析瓶頸：**針對 LLM 難以執行逆向 data-flow 分析的結構性弱點，設計專屬 Prompt 或生成機制予以補強。