AILS-II Enhanced: Automated Control Discovery via LLMs and NPU-Accelerated Cooperative Search

CVRPLib BKS Challenge — Team Registration Document

1 Team Information

Team Name: OptVerse-CityU

Members: Zhiwu An[1], Xin Chen[2], Qinglong Hu[2], Fei Liu[2], Mahdi Mostajabdaveh[1], Zidong Wang[2], Shunyu Yao[2], Kefeng Zheng[2], Zirui Zhou[1], Xialiang Tong[1], Mingxuan Yuan[1], Qingfu Zhang[2]

Affiliations: [1]Huawei Technologies, [2]City University of Hong Kong

Contact: anzhiwu1@huawei.com, xchen3252-c@my.cityu.edu.hk, qinglhu2-c@my.cityu.edu.hk, fliu36-c@my.cityu.edu.hk, mahdi.mostajabdaveh1@huawei.com, zidowang@cityu.edu.hk, shunyuyao8-c@my.cityu.edu.hk, kefezheng2-c@my.cityu.edu.hk, zirui.zhou@huawei.com, tongxialiang@huawei.com, yuan.mingxuan@huawei.com, qingfu.zhang@cityu.edu.hk

** Team members are listed in alphabetical order by surname.*

2 Abstract

We present a hybrid approach that extends AILS-II [5] along two complementary directions: (i) **LLM-guided adaptive control discovery** using Evolution-of-Heuristics (EoH) to automatically design improved diversification–intensification mechanisms, and (ii) **NPU-accelerated parallel search** with cooperative elite pool management. Our methods specifically target the *late-stage search regime* where discovering improvements over already high-quality solutions is the most challenging.

3 Proposed Methodology

3.1 Base Algorithm: AILS-II

We build upon AILS-II [5], leveraging its adaptive diversity control. Two parameters govern the diversification–intensification trade-off:

- **Perturbation degree (ω):** Dynamically adjusted (with adjustment frequency γ) based on the idea distance between the reference solution and the solution obtained after local search. The ideal distance gradually decreased during the algorithm to shift from diversification to intensification.
- **Acceptance criterion:** A convergent threshold-based criterion where solutions are accepted if their cost is below $\theta = f^{best} + \eta(\bar{f} - f^{best})$, with f^{best} being the best solution in the last γ iterations and $\bar{f}$ the average local search quality. The parameter $\eta \in [\eta_{min}, \eta_{max}]$ controls relaxation, starting at η_{max} (permissive) and converging to η_{min} (restrictive).

3.2 Direction 1: LLM-Guided Adaptive Control Discovery

Motivation. While AILS-II's hand-crafted adaptive rules achieve state-of-the-art performance, we hypothesize that automatically discovered control mechanisms can achieve superior late-stage convergence behavior. This is critical for the BKS Challenge, where initial solutions will already be of very high quality due to extensive runs by the organizers.

Method. We employ **Evolution-of-Heuristics (EoH)** [4], a framework that uses Large Language Models to generate, mutate, and evolve algorithmic components. To fully characterize the discovery process, we specify:

- **The representation of candidate mechanisms:** The LLM generates Java functions that take as input the current search state (iteration count, recent improvement history, current ω value, solution distance statistics) and output updated values for the perturbation degree ω and acceptance threshold.
- **The evaluation protocol:** Candidate control rules are evaluated using a *warm-start* protocol, initializing from known high-quality solutions (within 0.1–0.5% of BKS) rather than random starts. This is to ensure mechanisms are optimized for the "endgame" where marginal improvements matter most. Fitness is measured by improvement magnitude and consistency on training instances from the CVRPLib XL generator.
- **The evolutionary operators:** The LLM proposes novel functions (generation), modifies existing high-performing functions (mutation), and combines elements from multiple successful functions (crossover). Selection is based on fitness ranking across the candidate population.
- **The integration of discovered mechanisms:** Top-performing control functions replace the default AILS-II adaptive rules for ω and acceptance decisions. We aim for a general rule that can be use for all of the instances.

3.3 Direction 2: NPU-Accelerated Parallel AILS-II

Motivation. Recent advances in GPU-accelerated optimization (cuOpt [6], PDLP [1]) demonstrate that massive parallelism can yield order-of-magnitude speedups with better exploration. We use Huawei Ascend NPUs for parallelization.

Architecture. To fully characterize the cooperative parallel search, we specify:

- **The information shared:** A shared elite pool maintains the k-best solutions found across all workers. Additionally, we maintain *string frequency statistics*—counts of consecutive node sequences appearing in elite solutions.
- **The cooperative methods:** Two components operate concurrently: (A) *Parallel AILS workers* executing independent searches with different seeds; (B) *Path-relinking worker* [3] generating intermediate solutions between elite pairs and reinjecting improvements into the pool.
- **The communication timing:** Workers contribute to the elite pool immediately upon finding a new global best. Stagnating workers (no improvement for T iterations) trigger a restart. String frequencies are updated incrementally as the elite pool evolves.
- **The utilization of imported information:** Knowledge transfer among workers occurs through two mechanisms: (1) *Cooperative restart*—stagnating workers restart from a randomly selected elite solution, inheriting high-quality structure; (2) *Frequency-guided perturbation*—inspired by string-based methods [2], this perturbation operator exploits the shared frequency statistics to preserve high-frequency strings (likely high-quality partial structures) while targeting low-frequency edges for removal/reinsertion, biasing search toward promising regions.

3.4 Integration Strategy

The two directions are complementary: LLM-discovered adaptive controls govern individual worker behavior, while the parallel architecture provides computational scale and cooperative intensification. Configurations selected based on synthetic XL instance performance.

4 Third-Party Components and Acknowledgments

- **AILS-II** [5]: Base framework (from `https://github.com/INFORMSJoC/2023.0106`)
- **EoH Framework** [4]: LLM-guided heuristic evolution
- **CVRPLib XL Generator**: Training instances (organizers)

All novel components (control discovery, parallel architecture, frequency-guided perturbation) are original contributions.

References

[1] D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 20243–20257, 2021.

[2] J. Christiaens and G. Vanden Berghe. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433, 2020.

[3] F. Glover. Tabu search and adaptive memory programming—advances, applications and challenges. In R. S. Barr, R. V. Helgason, and J. L. Kennington, editors, *Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies*, pages 1–75. Kluwer Academic Publishers, Boston, 1997.

[4] F. Liu, X. Tong, M. Yuan, X. Lin, F. Luo, Z. Wang, Z. Lu, and Q. Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 32201–32223. PMLR, 2024.

[5] V. R. Máximo, J.-F. Cordeau, and M. C. V. Nascimento. AILS-II: An adaptive iterated local search heuristic for the large-scale capacitated vehicle routing problem. *INFORMS Journal on Computing*, 36(4):974–986, 2024.

[6] NVIDIA. cuOpt: GPU-accelerated solver for vehicle routing problems. `https://developer.nvidia.com/cuopt`, 2024. Accessed: 2024.