# Deep Learning based Cross-domain Optimization through Selection Hyper-heuristics

Zian Pan and Tzu-liang Huang
Supervised by Prof Mustafa Misir, Ph.D.
Supported by DKU Undergraduate Studies Office and the Summer Research Scholars Program

## Introduction & Motivation

**Algorithms Design** is an essential process for Search and Optimization tasks. But this task becomes tricky due to the large algorithm design space and traditional human-conceived algorithms are mostly problem-specific and always need great efforts of work. **Hyper Heuristics (HHs)**, on the other hand, is designed to embody the problem-independent trait, which means it can solve a number of problems using the same algorithm [1].

In our research, we focus on **Selection Hyper Heuristics (SHHs)**, one category belonging to **HHs**. **SHHs** can be interpreted as an algorithm that selects algorithms through **iterations**. The algorithms in the selecting space are called **Low Level Heuristics (LLHs)**, which are some naive algorithms that perform directly on the solution space. Since SHH involves making decisions on which **LLH** to choose, we want to implement **Reinforcement Learning (RL)** methods in this seleting process, as **RL** has shown its great effectiveness in computer games, autonomous cars and so forth. All the evaluations are tested on **Hyflex**, a java platform that provides several problems with their **LLH** set accordingly.

## Hyper Heuristic

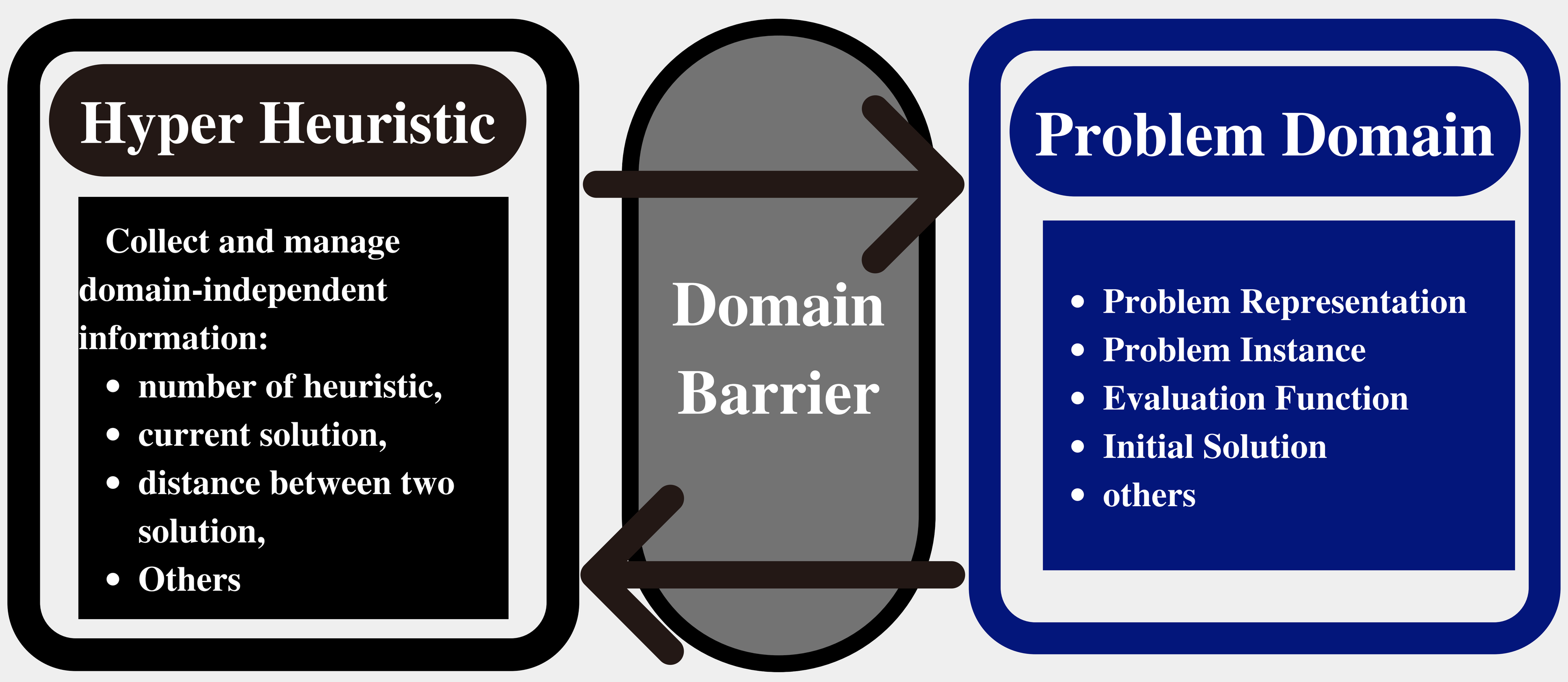


*Figure 1. A sturcture of Hyper Heuristic*

By utilizing the domain-independent information, **HH** algorithms can determine the right action to take without knowing any of the prior knowlegde of the problems.

## Reinforcement Learning

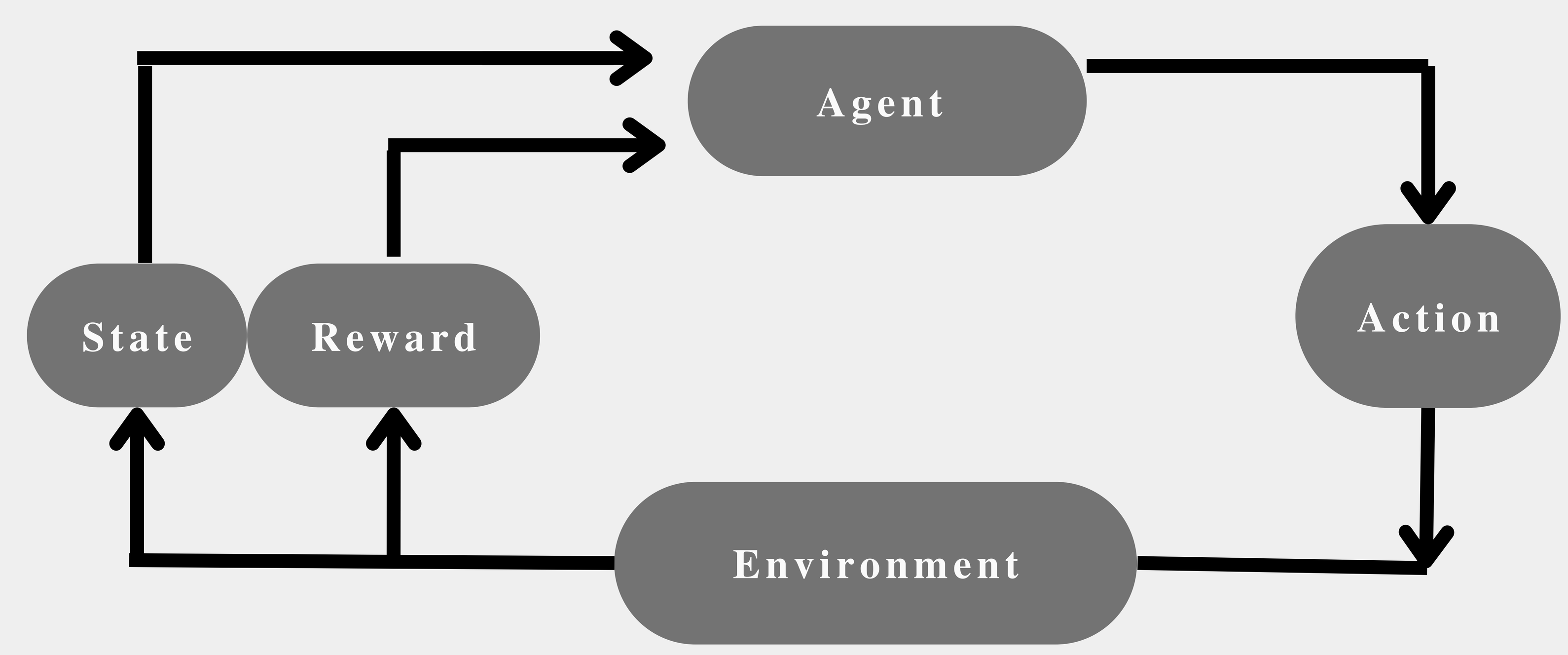


*Figure 2. Basic Reinforcement Leanring stucture*

**Reinforcement learning (RL)** is a powerful learning paradigm in machine learning and artificial intelligence. It is concerned with decision-making problems where the goal is to maximize the expectation of rewards by choosing the "right" action at each state.


*Figure 4. Performance Comparison between Q Learning with backrack and Random HH (Since the task is for minimization, the lower the value is, the better the performance is.)*

## Current Methods & Results

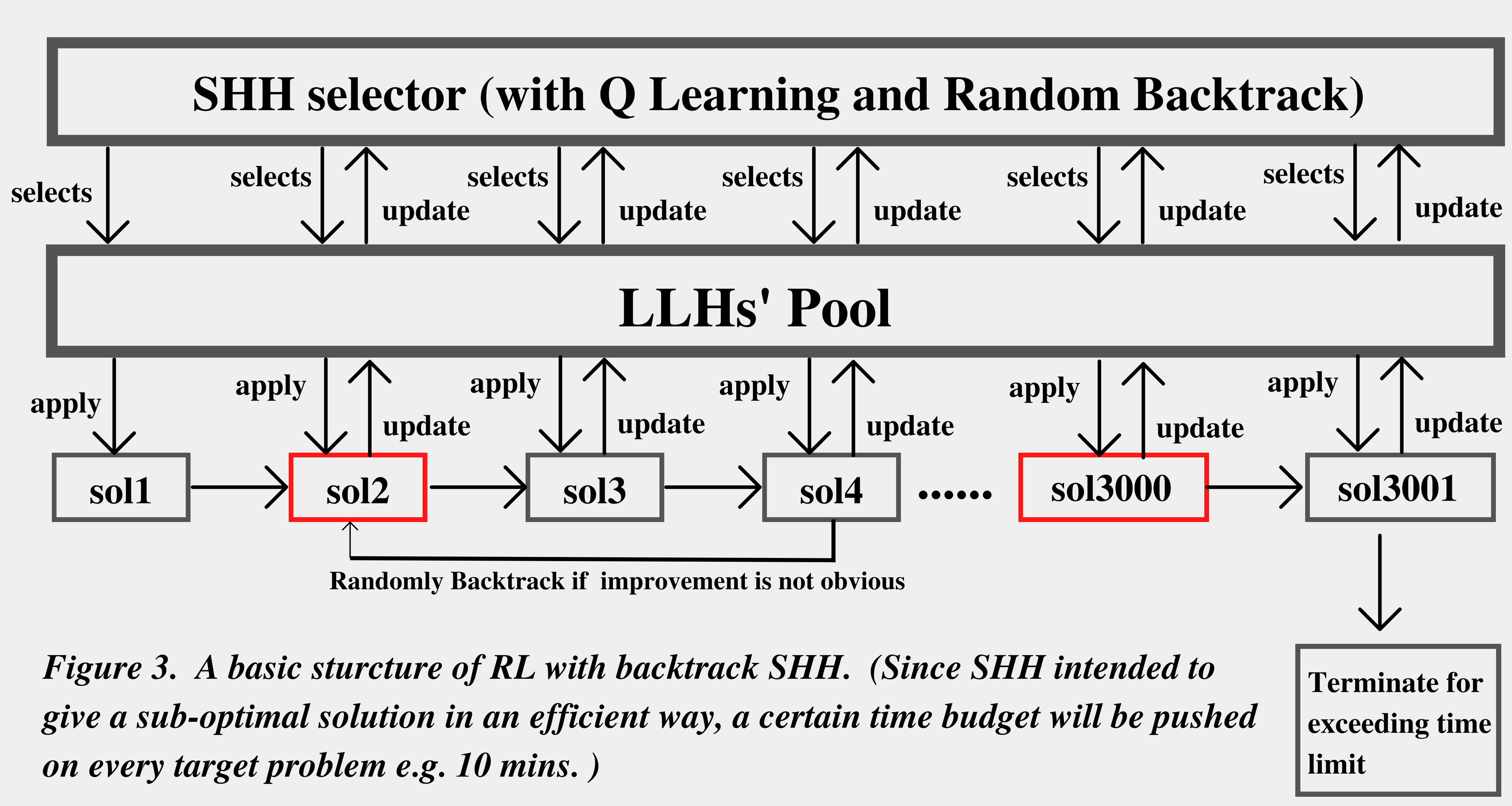


*Figure 3. A basic sturcture of RL with backtrack SHH. (Since SHH intended to give a sub-optimal solution in an efficient way, a certain time budget will be pushed on every target problem e.g. 10 mins. )*

### Methods

**Q Learning with Backtrack:**

**State:**

1. **Tail Length** (how many numbers of trails of LHHs are left in an episode) [2]
2. **Last Heuristic Applied** (the index of the last LHH applied) [2]

**Reward:**

Total improvement divided by the total time used in each episode

**Backtrack:** Every time the RL agent gets stuck and no improvements are found, it will backtrack and start from a previous solution (recorded randomly)

### Results

We tested our algorithms in the three problems, **Maximum Satisfiability Problem (MAX-SAT), Travelling Salesman Problem (TSP), and Bin Packing (BP)**, and compared the results with the Random Selection HH, which randomly chooses LLH to apply until the improvement is found. The results of our Q Learning with backtrack show significant improvement in performance.

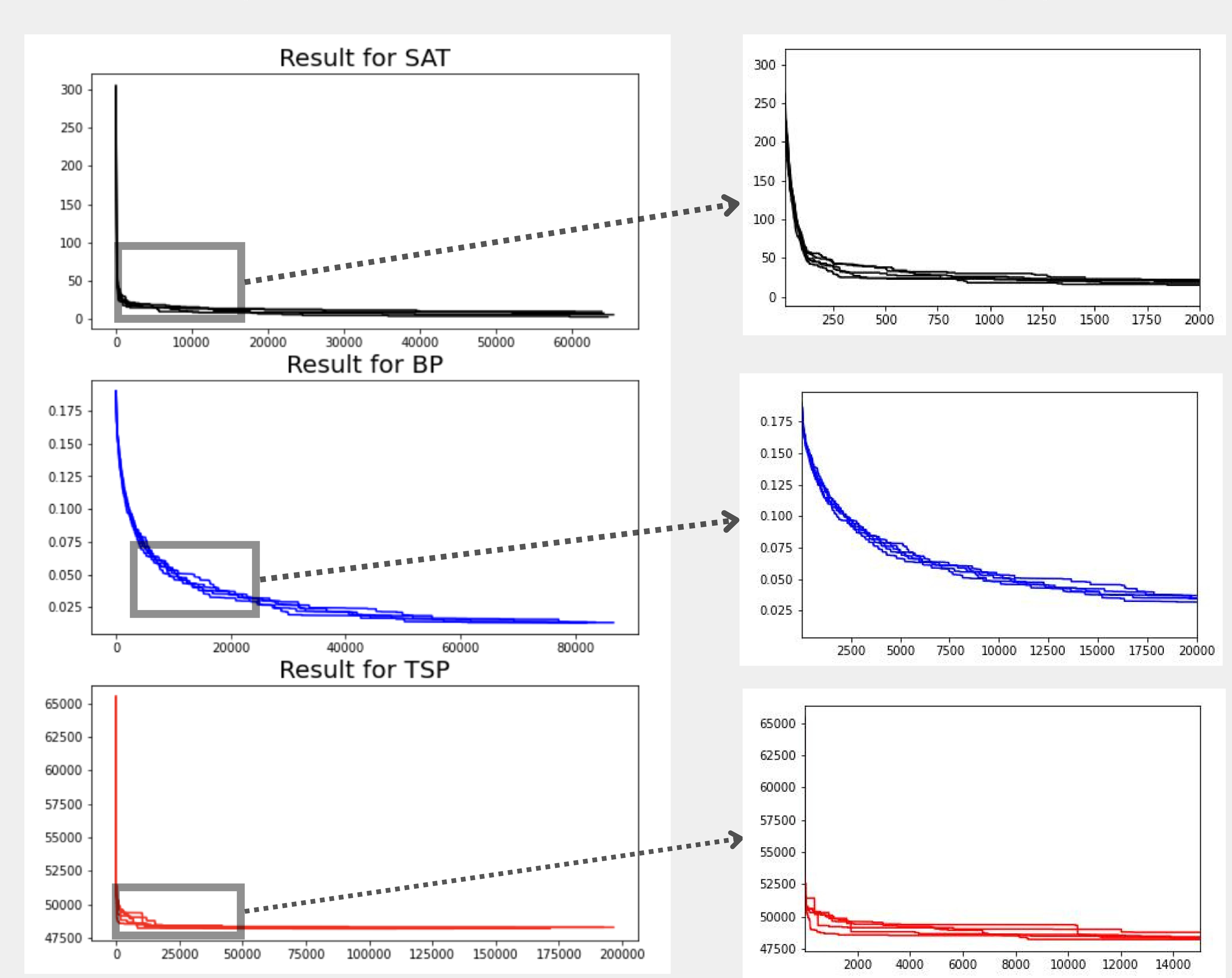


*Figure 5. The optimization process of Q Learning with backtrack performing on the three problems. Each line represents onr time of execution of our alogorithm. (5 executions for each problem)*

**Reference**
[1] M. Mısır, "Hyper-heuristics: Autonomous Problem Solvers," in Automated Design of Machine Learning and Search Algorithms, N. Pillay and R. Qu, Eds. Cham: Springer International Publishing, 2021, pp. 109–131. doi: 10.1007/978-3-030-72069-8_7.
[2] F. Mischek and N. Musliu, "Reinforcement Learning for Cross-Domain Hyper-Heuristics," in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria, Jul. 2022, pp. 4793–4799. doi: 10.24963/ijcai.2022/664.