Introduction

- Hyper-heuristics are Search techniques that:
  - Select among low-level heuristics, or
  - generate heuristics from components.
- Selection Hyper-heuristic consists of two main components:
  1. Selection mechanism.
  2. Move acceptance strategy.

Methodology

- The runtime of a simple selection hyper-heuristic using two learning mechanisms, namely Simple Random and Random Gradient, were analysed in a general scenario that can be described as follows:
  1. Given a finite search space \( S \), the aim is to maximise a fitness function \( f: S \rightarrow \mathbb{R} \), for example, to find \( x \in S \) and, such that \( f(x) = \max_{y \in S} f(y) \).
  2. We used the artificial fitness level technique.
  3. The search space is divided into a number of fitness-levels \( (S_1, S_2, \ldots, S_m) \), where the quality of solution in fitness-level \( S_{i+1} \) is better than that in fitness-level \( S_i \). The optimal solution is always in the last fitness-level \( S_m \).
  4. We assume that we have \( m \) low-level heuristics, and the success probability of any given heuristic is the same in each fitness-level.

Results

- **Theorem 1**: the expected runtime of Simple Random in the specified scenario is:
  \[
  \mathbb{E}[T] \leq m (n - 1) \frac{1}{\Sigma_{i=1}^{m} p(i)}
  \]
- **Theorem 2**: the expected runtime of Random Gradient in the specified scenario is less than or equal to:
  \[
  (n - 1) \frac{1 + \frac{m}{\Sigma_{i=1}^{m} p(i)}}{\Sigma_{i=1}^{m} (p(i))^2} \frac{1 - \frac{\frac{\Sigma_{i=1}^{m} (p(i))^2}{\Sigma_{i=1}^{m} p(i)}}{1}}{m}
  \]
- Random gradient outperforms simple random, no matter how large the problem instance is, if:
  \[
  \frac{\Sigma_{i=1}^{m} (p(i))^2}{\Sigma_{i=1}^{m} p(i)} < \frac{1}{m}
  \]
- The success probabilities of low-level heuristics play an important factor.

Conclusion

- Selection hyper-heuristics empirically shown to be successful.
- The theoretical background of selection hyper-heuristics is very weak.
- This research is one of the initial studies on the runtime analysis of selection hyper-heuristics.
- Runtime analysis provides insightful view of selection hyper-heuristics behaviour, which in turn gives a guideline for practitioners.

References