Hyper Metaheuristics

The term **hyper-heuristic** is relatively new; it was first used in 2000 to describe *heuristics to choose heuristics* in the context of combinatorial optimization.

The definition of hyper-heuristics has been recently extended to refer to a search method or learning mechanism for selecting or generating heuristics to solve computational search problems.

The distinguishing feature of hyper-heuristics is that they operate on a search space of heuristics (or heuristic components) rather than directly on the search space of solutions to the underlying problem that is being addressed.

References:
Burke et. al. (2014), "A Classification of Hyper-heuristic Approaches",
N. R. Sabar and G. Kendall, "Population based Monte Carlo tree search hyper-heuristic for combinatorial optimization problems," Information Sciences, vol. 314, pp. 225-239, 2015.
Nugraheni and Abednego (2014), "A Combined Meta-Heuristic with Hyper-Heuristic Approach to Single Machine Production Scheduling Problem".

Hyper Metaheuristics

Despite the success of heuristic methods and other search techniques in solving real-world computational search problems, there are still some difficulties in terms of easily applying them to newly encountered problems, or even new instances of similar problems. These difficulties arise mainly from:

The significant range of parameter or algorithm choices involved when using this type of approach and the lack of guidance as to how to select them	different heuristics impose different strength and weakness. Thus, it makes sense to merge them into one framework	the scientific community's level of understanding of why different heuristics work effectively (or not) in different situations does not facilitate simple choices of which approach to use in which situation

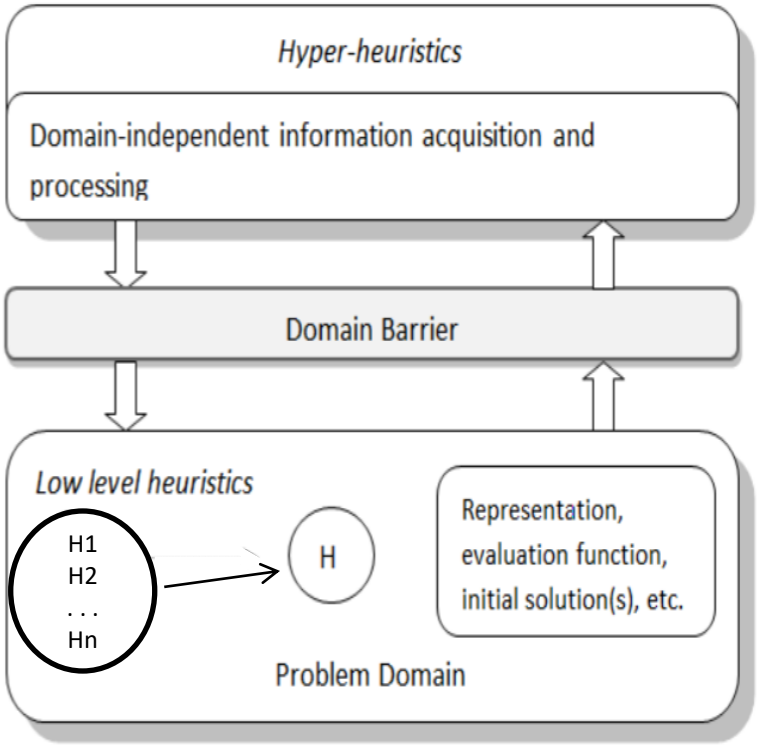
So, a key motivating goal for this area is the challenge of automating the design and tuning of heuristic methods to solve hard computational search problems.

Hyper Metaheuristics

A generic hyper-heuristic framework is composed of two levels known as high-level and low-level heuristics.

The high-level heuristic is problem independent and has no domain knowledge. Its role is to manage the selection or generation of which heuristic are to be applied at each decision point.

The low-level heuristic corresponds to a pool of heuristics or heuristic components.



Hyper Metaheuristics

When using hyper-heuristics, we are attempting to find the right method or sequence of heuristics in a given situation **rather than trying to solve the problem directly**.

A hyper-heuristic can be seen as a (high-level) methodology which, when given a particular problem instance or class of instances, and a number of low-level heuristics (or its components), automatically produces an adequate combination of the provided components to effectively solve the given problem(s).

The classification of hyper-heuristic approaches considers two dimensions:

1) the nature of the heuristics' search space. Here, we have:

heuristic selection: methodologies for choosing or selecting existing heuristics.	heuristic generation: methodologies for generating new heuristics from the components of existing ones.

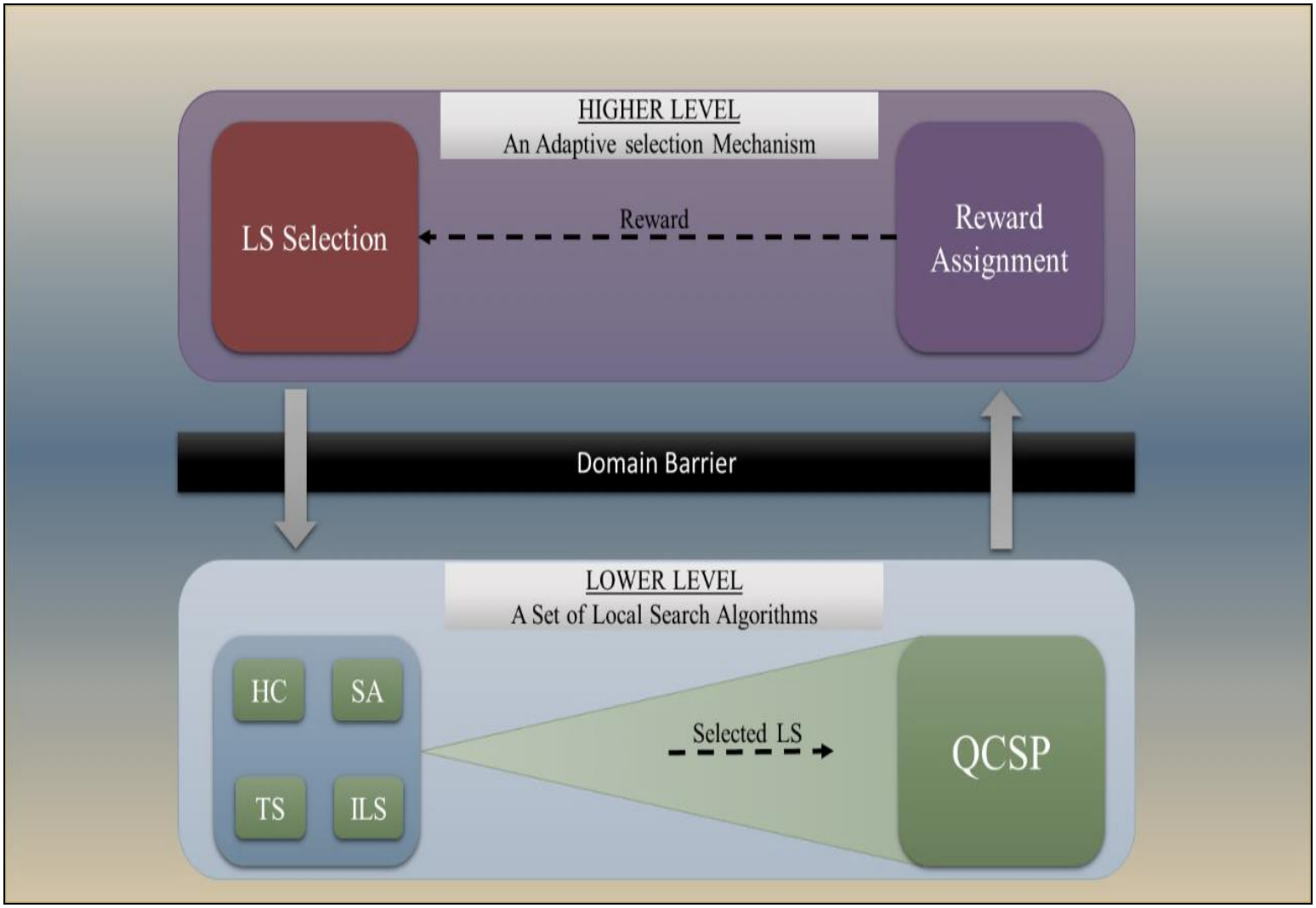
The distinction between constructive and perturbative search paradigms.

2) the different sources of feedback information

A hyper-heuristic is a learning algorithm when it uses some feedback from the search process.

- Online learning: the learning takes place while the algorithm is solving an instance of a problem
- Offline learning: gather knowledge in the form of rules or programs, from a set of training instances, that will hopefully generalise to solving unseen instances.

No-learning: Do not use feedback from the search process.


HIGHER LEVEL
An Adaptive selection Mechanism
LS Selection
Reward
Reward Assignment
Domain Barrier
LOWER LEVEL
A Set of Local Search Algorithms
HC
SA
TS
ILS
Selected LS
QCSP

THANK YOU