Guided Local Search

Q: Heuristic/Metaheuristics
A ***heuristic*** technique, often called simply a heuristic, is any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals.

In computer science and mathematical optimization, a **metaheuristic** is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity. Metaheuristics sample a set of solutions which is too large to be completely sampled. Metaheuristics may make few assumptions about the optimization problem being solved, and so they may be usable for a variety of problems.
Metaheuristic algorithms are approximate and usually non-deterministic.

Local Search Strategy (Metaheuristics)
A well known local search algorithm is the hill climbing method which is used to find local optimums. However, hill climbing does not guarantee finding global optimum solutions.

Many metaheuristic ideas were proposed to improve local search heuristic in order to find better solutions. Such metaheuristics include simulated annealing, tabu search, iterated local search, variable neighborhood search, and GRASP. These metaheuristics can both be classified as local search-based or global search metaheuristics.

Other global search metaheuristics that are not local search-based are usually population-based metaheuristics. Such metaheuristics include ant colony optimization, evolutionary computation, particle swarm optimization, and genetic algorithms.

Guided Local Search is a metaheuristic search method. A meta-heuristic method is a method that sits on top of a local search algorithm to change its behavior.

Guided Local Search builds up penalties during a search. It uses penalties to help local search algorithms escape from local minimal and plateaus. When the given local search algorithm settles in a local optimum, GLS modifies the objective function using a specific scheme (explained below). Then the local search will operate using an augmented objective function, which is designed to bring the search out of the local optimum. The key is in the way that the objective function is modified.

The problem is usually expressed as follows:

Given a set of customers requiring a visit, and a fleet of vehicles based at a depot that can perform the visits, construct a set of routes for the vehicles which minimises the costs of operation.
The objective function is usually expressed as costs related to the number of vehicles used and to distance travelled.

The Traveling Salesman Problem (TSP) is one of the most famous problems in combinatorial optimization. In this paper, we are going to examine how the techniques of Guided Local Search (GLS) and Fast Local Search (FLS) can be applied to the problem. GLS sits on top of local search heuristics and has as a main aim to guide these procedures in exploring efficiently and effectively the vast search spaces of combinatorial optimization problems.

Combinatorial Optimisation: In operations research, applied mathematics and theoretical computer science, combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects. In many such problems, exhaustive search is not tractable.

Computational Complexity:
P - Problem solvable in polynomial time
EXP - Problem solvable in exponential time
Note: P is a subset of EXP

R - Problem solvable in finite time (Solvable and computable problems).

NP(Non-deterministic Polynomial)

- A problem is assigned to the NP (nondeterministic polynomial time) class if it is solvable in polynomial time by a nondeterministic Turing machine.
- Decision Problems solvable in polynomial time via a “Lucky Algorithm”.
- Nondeterministic model (algorithm makes guesses).

Another way to think about NP:
Decision problems with solutions that can be “checked” in polynomial time.

- When answer = yes, can prove it & check time in polynomial time.

NP Hard - It’s at least as hard as every polynomial in NP

NP Complete - NP and NP Hard