**CS4234: Optimisation Algorithms** Tutorial 10

# SLS Part 2; Final Preparation 1

*V1.4: Steven Halim* *November 2, 2020*

## Preliminaries

During Lecture 9, 10, and the previous Tutorial 9, you have been exposed with a new search paradigm: (Stochastic) Local Search (SLS) followed by 4 (FOUR) more established SLS algorithms (also called 'Meta-heuristics'): SA, TS, ILS, MA, that are reported to be successful in attacking various (NP-)hard COPs $A$, $B$, $C$, ... in various research papers. In this tutorial, we will discuss various statements about SLS.

## Discussion Points

**Q1: Statements About SLS (up to Lecture 10)** For each statement below about Stochastic Local Search (SLS) algorithm, determine if it is More Towards True/More Towards False/It depends and give a short explanation.

1. We can run an SLS algorithm (the first 'S' = Stochastic) for an NP-hard Combinatorial Optimization Problem (COP) instance for an **extremely long time**, e.g., $\approx \infty$, and still unable to prove that the best found solution of that run is the Global Optima (GO) for that COP instance.

2. **All** SLS algorithms, if run for **extremely long time**, e.g., $\approx \infty$, will **always** encounter a GO of a COP instance during its long search run although it cannot stop immediately after encountering such GO (see the previous statement).

3. SLS algorithms that use larger neighborhood is **always** better than SLS algorithms that use smaller neighborhood.

4. It may be possible to provide an approximation ratio for an SLS algorithm even when we only run the SLS algorithm for a finite amount of time. Current Computer Scientists are just not yet able to prove the approximation ratio of an SLS algorithm yet.

5. Hybrid SLS algorithms (that combines two, or more, simpler SLS algorithms) is **always better** than its individual SLS algorithm working individually on its own.

6. Tabu Search (TS) algorithm is a better SLS algorithm than Simulated Annealing (SA).

7. We can make *any SLS algorithm* for Metric No-Repeat TSP to have a 2-approximation ratio.

8. In Tabu Search algorithm, setting high Tabu Tenure value/setting encourages diversification search strategy.

9. If we use Tabu Search for TSP, the best parameter setting for Tabu Tenure is a fixed constant 7, i.e., that is, forbid the last 7 local moves that Tabu Search has just performed.

**Q2:** Past paper (AY2019/20) hidden MCQs:

1. Which statement about Stochastic Local Search (SLS) algorithm is correct?
   (a) Albeit more difficult, we can analyze the worst case time complexity of an SLS algorithm
   (b) SLS algorithm terminates upon finding Global Optima
   (c) We should use SLS algorithm when we are given an NP-hard optimization problem
   (d) We can do pre-processing to make any SLS algorithm for (M-NR-)TSP has 2-approximation bound
   (e) It is easy to design a good SLS algorithm for a given NP-hard optimization problem
2. Which statement about Tabu Search is incorrect?
   (a) It has an optional component called Aspiration Criteria
   (b) It uses cooling function
   (c) One of its most important tunable parameter is Tabu Tenure
   (d) Lowering Tabu Tenure value makes Tabu Search perform more intensification
   (e) It is more efficient to forbid recent local moves instead of recently found solutions
3. Which of the following animal-inspired SLS/metaheuristic has never appeared in at least one scientific article before?
   (a) Ants Colony Optimization
   (b) Bat algorithm
   (c) Cuttlefish optimization algorithm
   (d) Killer Whale algorithm
   (e) Actually, all metaheuristics a-d above have appeared in at least one scientific article before

**Q3:** Final Assessment Preparation (past Kattis problems). As you might have suspected, Steven usually put some (NP-)hard problems from past programming competitions in his paper. So, try to solve the following problems at Kattis that have been used in CS4234/CS3233 before. Your TA will discuss a few (possibly not all) problems:

- `https://nus.kattis.com/problems/piano` (used in S1 AY2017/18)
- `https://nus.kattis.com/problems/hidingchickens` (used in S1 AY2017/18)