CS4234: Optimisation Algorithms Tutorial 9

Stochastic-Local-Search/Meta-heuristics

V1.2: Steven Halim *November 05, 2018*

1 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. Quadratic Assignment Problem (QAP) (also see Section 2 below) is quite close to Traveling Salesman Problem (TSP). In fact, most NP-hardness proof of QAP is derived from reducing NP-hard TSP into QAP. Thus, *any SLS algorithm* that empirically works well for TSP should also works well for QAP.

10. 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.

2 Delta Evaluation for QAP

The delta evaluation technique has been shown to be very useful for achieving good-performing SLS algorithms. The idea is generic enough: To compute the objective value of a neighboring candidate solution, one does not need to recompute from scratch by definition (usually with higher time complexity) but rather compute the delta changes instead (usually with much lower time complexity) as Local Search, as the name implies, only does local changes to the candidate solution.

In class, we have discussed $O(1)$ delta evaluation of 2-opt swap edges local move for TRAVELLING-SALESMAN-PROBLEM (current objective value - 2 deleted edges + 2 added edges) compared to $O(n)$ computation from scratch. You have also been exposed to $O(n)$ delta evaluation of 1-bit flip local move for LOW-AUTOCORRELATION-BINARY-SEQUENCE Problem (by analyzing the effect of that bit flip to values of $C(k)$) compared to $O(n^2)$ computation from scratch.

Let's briefly discussed another NP-hard COP: QUADRATIC-ASSIGNMENT-PROBLEM (QAP). The formal problem description of QAP is as follows: Given two $n \times n$ matrices $A = (a_{ij})$ (typically contains flow information between facilities) and $B = (b_{ij})$ (typically contains distance information between facilities), find an assignment (a permutation) s of $\{0, 1, 2, ..., n\text{-}1\}$ over all possible permutations in the search space which minimizes the objective function $g(s) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_{s_i s_j} \times b_{ij}$. As you can see, computation from scratch is clearly $O(n^2)$. Now, please design a faster-than-$O(n^2)$ delta evaluation for swap 2 locations local move for QAP analyze its time complexity!

2.1 Manual Computation

Please use this example to help you design a fast delta evaluation.
You are given two 4×4 matrices A (left) and B (right) below:

0	8	3	2
0	0	0	1
0	2	0	0
2	0	0	0

0	1	8	9
1	0	2	1
8	2	0	5
9	1	5	0

Now, assuming that $s = \{0, 1, 2, 3\}$, compute $g(s)$:

Then, swap the first two facilities, i.e. we now have $s' = \{1, 0, 2, 3\}$ now and compute $g(s')$:

What if we swap the middle two facilities, i.e. we have $s'' = \{0, 2, 1, 3\}$; compute $g(s'')$:

2.2 The Actual Delta Evaluation

Now based on the manual computation that you have performed earlier (you can do more), design (any) delta evaluation for swap 2 facilities local move for QAP that is faster-than-$O(n^2)$!