Proposal

The Quadratic Assignment Problem (QAP) is an NP-Hard combinatorial optimization problem where the major goal is to assign a set of facilities to locations in an optimal manner. Each set of facilities has a corresponding flow, while each set of locations contains a distance. The goal is to find an optimal permutation such that when the optimal permutation of facilities is assigned to locations, the sum of products between flows and distances is minimal. More formally, given two sets C = [pij] and D=[dij] find the permutation π\* ∈ Π such that is minimal. This problem is well-known not only due to its difficulty in solving even the smallest of problem sets but due to its inherent encapsulation of many real-life complex problems such as backboard wiring, Campus and Hospital Layout, Typewriter keyboard design, and Scheduling.

The Guided Local Search (GLS) is a heuristic recently developed for the problem that when equipped with a local searching mechanism is able to produce excellent results on some of the hardest problem instances while producing lackluster results on other more structure problem instances of QAP. The general precept of the guided local search in order to allow traversals past local minimum is to inject noise into the objective function at certain features of currently seen solutions. Applying a hill-climbing local search mechanism then to the adjusted landscape then will seek out new novel solutions while at the same time preventing traversal to previously-seen solutions, bypassing local optimum, and augmenting the search for the global optimum. Currently GLS performs outstandingly and surpasses the state-of-the-art algorithms: Robust Tabu Search (RoTS) and its more recent derivative Reactive Tabu Search (ReTS), on the TAI-A type of instances. Because in practice these are the hardest instances to solve due to their uniformly random generation and therefore wild unadaptable solution landscapes, the fact GLS has been able to outperform both RoTS and ReTS obtaining nearly twice better results, it shows GLS is a hidden gem of a heuristic that with proper modification can produce excellent results on Tai-B instances as well (structured instances non-uniformly generated to model real-life instances).   
  
Research Our main research plans include adjusting the Guided Local Search to equip it with more methods for diversification by adopting the idea from RoTS and ReTS – namely, that of the aspiration criteria. This will enforce a certain swap to occur if the swap hasn’t occurred for the past swaps (iterations). This seems far more intuitive and productive a means to diversification than simply enforcing random swaps to occur some percent of the time. Even the more naïve random swapping mechanism allowed GLS to perform much better than simply hill-climbing a dynamic augmented function (or a noise-modified solution landscape, if you will). We also plan to investigate the ant-colony optimization notion of pheromone evaporation as it applies to GLS. Hence, instead of simply continuing to inject more and more noise into the landscape, we intend to slowly but constantly evaporate the modified landscape such that it slowly reforms into the original landscape overtime had no further noise injection taken place.

Goal

Our goals are ambitious. We plan to create a modified GLS that will be established as one of the state-of-the art meta-heuristics for conquering QAP. However, should all our attempts fail, we desire to minimally produce results better than the original GLS, and hopefully, at least better than RoTS in Taillard-B instances, since it is already better in Taillard-A instances. The reason for focus on these two problem types are that these two types well survey the major problem types of QAP.

# Testing

We will be testing on the excellent set of benchmarks supplied by QAPLib. We will test on 10 problem instances surveying the entirety of all the 4 problem types of QAP, however we will focus our efforts on Tai-A and Tai-B instances. Furthermore, all runs will be performed simultaneously on GLS, our modified GLS, and RoTS. This will provide a test-bed for comparison between the algorithms. We may optionally perform testing on ReTS if we can find code available on the internet for the algorithm. Finally, we will minimally run each algorithm 5-10 times on each problem instance and display only the averaged results along with standard deviations for a more robust comparison.