To speed up the implementation of 2-opt algorithm, fixed-radius search with candidate lists can be used.

candidate lists

In the context of identifying candidates for k -exchange moves, it is useful to be able to efficiently access the vertices in the given graph G that are connected to a given vertex u_i by edges with low weight, for example, in the form of a list of neighbouring vertices u_k that is sorted according to edge weight w ((u_i , u_k)) in ascending order. By using such candidate lists for all vertices in G , fixed radius near neighbour searches can be performed very efficiently.

fixed-radius search

For any improving 2-exchange move from a tour s to a neighbouring tour s' , there is at least one vertex that is incident to an edge e in s that is replaced by a different edge e' with lower weight than e . This observation can be exploited for speeding up the search for an improving 2-exchange move from a given tour s .

For a vertex u_i , two searches are performed that consider each of the two tour neighbours of u_i as a vertex u_j , respectively. For a given u_j , a search around u_i is performed for vertices u_k that are closer to ui than w ((ui , uj)), the radius of the search. For each vertex u_k found in this fixed radius near neighbour search, removing one of its two incident edges in s leads to a feasible 2-exchange move. The first such 2-exchange move that results in an improvement in solution quality is applied to s , and the iterative improvement search is continued from the resulting tour s' by performing a fixed radius near neighbour search for another vertex. If fixed radius near neighbour searches for all vertices do not result in any improving 2-exchange move, the current tour is 2-optimal.

bounded candidate lists

It is often preferable to use bounded-length candidate lists; in this case, a fixed radius near neighbour search for a given vertex u_i is aborted when the candidate list for u_i has been completely examined, if the radius criterion did not stop the search earlier. As a consequence, the tours obtained from an iterative improvement algorithm based on this mechanism are no longer guaranteed to be locally optimal, because some improving moves may be missed.

Usually, candidate lists of length 10 to 40 are used.