Nearest-neighbor heuristic

Starting from some point p0, we walk first to its nearest neighbor p1. From p1, we walk to its nearest unvisited neighbor thus excluding only p0 as a candidate. We now repeat this process until we run out of unlisted points after which we return to p0 to close off the tour

NearestNeighbor(P)
    Pick and visit an initial point p_0 from P
    $p = p_0$
    $i = 0$
    While there are still unvisited points
        $i = i + 1$
        Select p_i to be the closest unvisited point to p_{i-1}
        Visit p_i
    Return to p_0 from p_{n-1}

ClosestPair(P)
    Let n be the number of points in set P.
    For $i = 1$ to $n - 1$ do
        $d = \infty$
        For each pair of endpoints (s, t) from distinct vertex chains
            if $dist(s, t) \leq d$ then $s_m = s$, $t_m = t$, and $d = dist(s, t)$
        Connect (s_m, t_m) by an edge
    Connect the two endpoints by an edge

Travel sales man problem

OptimalTSP(P)
    $d = \infty$
    For each of the $n!$ permutations P_i of point set P
        If $(cost(P_i) \leq d)$ then $d = cost(P_i)$ and $P_{min} = P_i$
    Return P_{min}

There is a fundamental difference between algorithms which always proceed a correct result and heuristics which may usually do a good job but without providing any guarantee