```
procedure OptPathPlan(graph G, path_set Q){
    while (G ≠ null){
        S = graphSplit(G); // split G into multiple disconnected graphs via DFS
        T = graphSelect(S); // select graphs not containing odd vertex from S
        if (T ≠ ∅){
            if (Q ≠ ∅){
                foreach (graph g in T){
                    T_path = findRelated(Q, g); // find a path having at least one g's vertex from Q
                    Q = Q – T_path; // remove T_path from Q
                    T_circuit = eulerCircuit(g); // return the Euler circuit of g as a new path
                    path = pathPaste(T_circuit, T_path); // connect path T_circuit with path T_path
                    Q = Q + path; // add path into Q
                    G = G - T_circuit; // delete the edges along path T_circuit from G
                }
            }
            else{
                foreach (graph g in T){
                    T_circuit = eulerCircuit(g); // return the Euler circuit of g as a new path
                    G = G - T_circuit; // delete the edges along path T_circuit from G
                    Q = Q + T_circuit; // add T_circuit into Q
                }
            }
        }
        foreach (graph g in S - T){ // for each graph containing odd vertex from S
            odd_num = getOddNum(g); // calculate the number of odd vertices in g
            if (odd_num == 2){ // there should be one Euler trail covering all edges of graph g
                path = eulerTrail(g); // find an Euler trail between the two odd vertices
            }
            else if (odd_num > 2){
                path = randomTrail(g); // find a trail between two randomly selected odd vertices
            }
            G = G - path; // delete the edges along path path from G
            Q = Q + path; // add path into Q
        }
    }
}
```