number of requests

Pickup locations

Store locations

Delivery locations

Set of all nodes. represents start and end node.

Set of vehicles

Capacity of vehicle

amount of pickup at location

time window of node

duration of service time at node

time cost of traveling from node to

1 iff the vehicle goes straight from to

when vehicle starts visiting

the load of vehicle after visiting

the time from to vehicle reach node

penalty when delayed delivery

Formulation

Objective function

st.

Tabu search

Algorithm 1

1. Generate initial solution s with adjusted cost

. Set it to be

1. iteration=0, Iteration\_total =0, newbest=0, initialize
2. While iteration<
   1. If newbest=1 or Iteration\_total %10==0 then

Intra-route exchange: for every vertex , sequentially remove its current route and reinsert in the best position to minimize

Set newbest=0

* 1. else

Inter-route exchange

* 1. If and s is feasible

, newbest=1, reset tabu list, iteration=0

Iteration++

* 1. Iteration\_total++

Algorithm 2 inter-route exchange

When are removed, predecessor and successor are connected. Then insert into another route , with existed order of vertices unchanged.

1. Search over neighborhood for the best solution, evaluated by f(s).
2. Update by . If respective constraints are not violated, divide by , otherwise, multiply by

Simulation：订单生成

For pickup locations , each one have random demand (Poisson distribution).

For each request, generate random

模式对比：

驻店：

每个骑手负责一家店。

每个骑手目标位最短时间内送完该店的所有订单。

满足

抢单：

1. 初始：各车辆按照次序选择时间最短的单，循环 次，并计算到达第一个送货点的时间
2. 车辆K排序（到达下一个送货地点的时间，当前订单数）
3. 选择rank1的车辆k
4. 计算车辆k接所有可接订单后增加的时间，让k接时间增加最短的单
5. 更新所有车辆到达下一个送货点的时间、订单数

每次选择时约束条件：

在满足约束条件下，每个车辆选择总时间最短的路径。（或者greedy，总是去距离当前位置最近的点。）