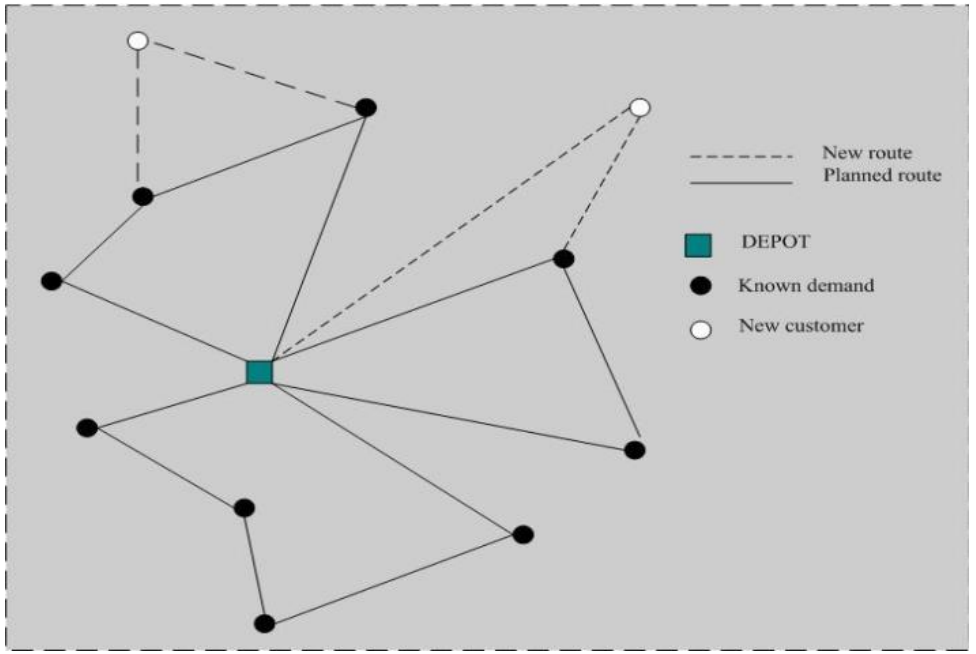
Dynamic pickup and delivery problem survey



Reference

[1] Berbeglia, Gerardo, Jean-François Cordeau, and Gilbert Laporte. "Dynamic pickup and delivery problems." European journal of operational research 202.1 (2010): 8-15.
[2] Psaraftis, Harilaos N., Min Wen, and Christos A. Kontovas. "Dynamic vehicle routing problems: Three decades and counting." Networks 67.1 (2016): 3-31.
[3] Gendreau, Michel, et al. "Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries." Transportation Research Part C: Emerging Technologies 14.3 (2006): 157-174
[4] Goel, Asvin, and Volker Gruhn. "Large Neighborhood Search for rich VRP with multiple pickup and delivery locations." (2005).
[5] Euchi, Jalel, Adnan Yassine, and Habib Chabchoub. "The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach." Swarm and Evolutionary Computation 21 (2015): 41-53.
[6]刘明德. 基于群体智能的动态需求车辆路径规划[D].哈尔滨工业大学,2020.

Dynamic pickup and delivery problem survey

Classfication：

- many-to-many（any vertex can serve as a source or as a destination for any commodity）
- one-to-many-to-one（eg. milk run）
- one-to-one(every commodity has a origin-destination pair)

Dynamic pickup and delivery problem survey

Objective function：

1、dynamic, such as average per unit time serviced customers, average per unit time cost, average demand rejections per unit time, or similar

2、static, such as route cost, total lateness, route distance

Dynamic pickup and delivery problem survey

Solution method

- Re-optimization

1、restart optimization based the current information (time consuming!!)

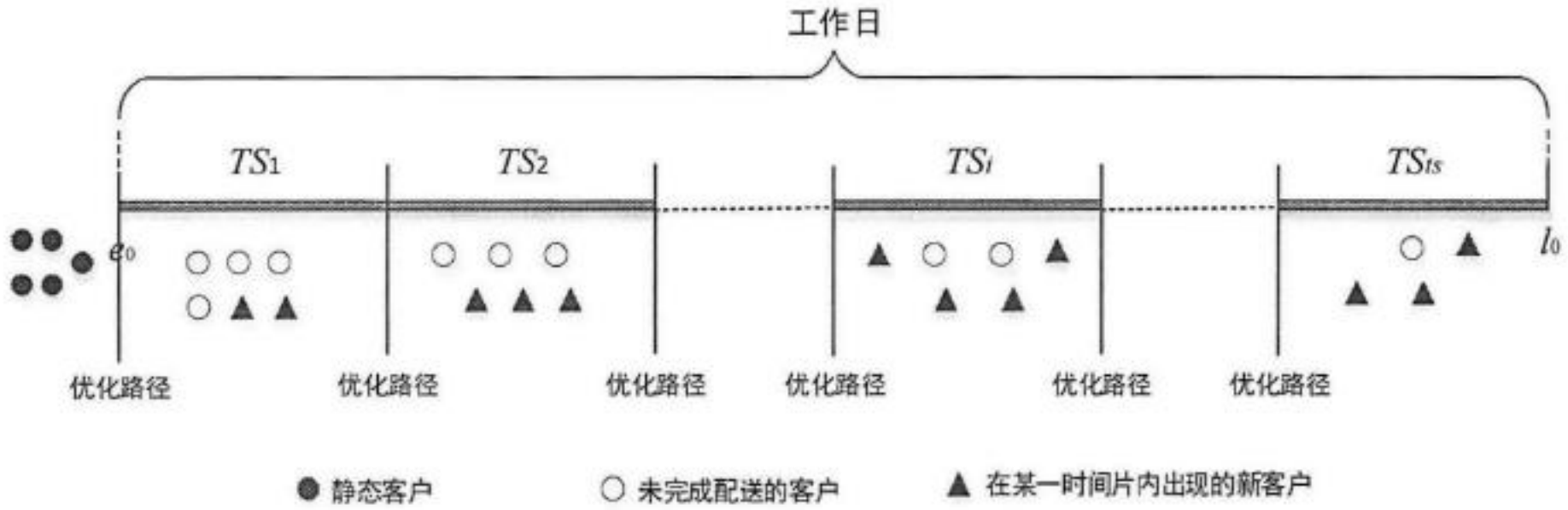
2、optimization based current solution(general, eg, insertion method)

Dynamic pickup and delivery problem survey

Solution method

- Strategy on dynamic request

1、periodic optimization strategy[4,5,6](general method, each time slice represent a static VRP)



Dynamic pickup and delivery problem survey

Solution method

- Strategy on dynamic demand

2、Event-driven optimization strategy[3](events such as new requests or requests cancellations , parallel computation)

1. while "no event" run the optimization procedure;
2. if "event" then
   (a) stop the optimization procedure;
   (b) if the event is "occurrence of a new request" then
      (i) update the solution found by the optimization procedure, as well as all solutions in the adaptive memory, through the insertion of the new request;
      (ii) add the updated solution found by the optimization procedure to the adaptive memory, if indicated;
      (iii) apply a local descent to the best solution in memory;
      otherwise ("end of service at a location")
      (i) add the solution found by the optimization procedure to the adaptive memory, if indicated;
      (ii) identify the driver's next destination, using the best solution stored in the adaptive memory;
      (iii) update all solutions in memory (see below);
   (c) restart the optimization procedure with a new solution generated from the adaptive memory.