PDP问题符号描述：

取货节点 pickup nodes：P = {1,...,n}

送货节点 delivery nodes：D = {n+1,...,2n}

|  |  |
| --- | --- |
| 符号 | 解释 |
|  | 需求点个数 |
|  | 车辆数 |
|  | 基于图的取货节点，{1,...,n} |
|  | 基于图的送货节点，{n+1,...,2n} |
|  | 车辆集合，|K|=m |
|  | 是能够满足需求的车辆集合，and 且 |
|  | and |
|  | ，开始端节点 |
|  | ，结束端节点 |
|  |  |
|  | 所有节点： |
|  | 所有弧边： |
|  | 对每一辆车，有子图： |
|  |  |
|  |  |
|  | ，，距离，非负 |
|  | ，，旅行时间，非负 |
|  | 节点的服务时间，，装卸时间 |
|  | 节点的时间窗，，可以提前到达不可提前访问 |
|  | 节点必须装到车上的货物，  ，  ， |
|  | 车辆容量， |

|  |  |
| --- | --- |
| 决策变量 | 解释 |
|  | =1，边使用车辆（） |
|  | 车辆从节点开始服务， |
|  | 服务节点离开后的货物装载上界， |
|  | =1，节点是为需求点 |

1. 时间满足三角不等式

，all for 

1. 假设，消除子回路和交货前的取货约束，进而更好的建模



目标函数：

（1）加权后最小化：行驶距离的总和，花费时间总和和每辆车的请求数量。



约束条件：

（2）确保每个拾取节点已被访问或者相应的请求已经请求银行。

 

1. 确保是同一辆车进行取送货。

 

1. 车辆离开每个起点的终端。

 

1. 车辆到达每个终点的终端。

 

1. 保证介于开始节点和终端节点之间是连续路径。

 

（7）保证车辆在节点服务遵守时间窗

 

（8）保证车辆在节点服务遵守时间窗

 

1. 保证取货在送货之前

 

1. 保证车辆的容量限制

 

1. 保证车辆的容量限制

 

1. 保证车辆的容量限制

 