**PDL Carbon-Neutral Festival and Event Logistics**

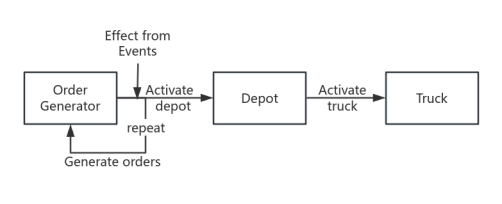
Attributes: TruckLocation, OutDepotTime, ActiveTime, DeliveryArrivalTime, InDepotTime,

The system under analysis is the logistics of a Festival and Event with the goal of Carbon-Neutral. This system aims to optimize energy consumption by managing truck deliveries and operations efficiently during large events. The key elements of the system include trucks, depots, service centers, orders, roads, event effect generators, and order generators. The system is modeled to simulate the logistics operations, incorporating dynamic elements like congestion from events and real-time adjustments in truck and order management.

System: A logistics simulation system based on the central city of Amsterdam during the King's Day.

Tracking energy usage per distance and time for trucks; optimizing route planning to reduce consumption; adjusting operations based on real-time congestion data generated by events.

**Research goal:** energy consumption reduction



**Analysis of per component**

**TTruck**

Attributes: TruckLocation, OutDepotTime, ActiveTime, DeliveryArrivalTime, InDepotTime, DistanceEnergyConsumption, TimeEnergyConsumption, NrOrder, DepotBelonging

Process: wait for service, deliver goods, plan shortest route, return to depot

**TDepot**

Attributes: DepotLocation, DepotID, NrInStockTrucks, OrderInStock, MaxOrder

Process: accept order, receive trucks

**TServiceCentre**

Attributes: NrServiceCapacity, ServiceTime, ServiceQueue,

Process: serve the trucks in the queue //including maintenance, loading and unloading goods and all preparation before delivery

**TOrder**

Attributes: OrderTime, TimeWindow, Destination, Volume

Process: none

**TRoad**

Attributes: RoadLocation, RoadID, VelocityLimit, Length

Process: none

**TEventEffectGenerator**

Attributes: VenueLocation, EventStartTime, EventEndTime, InfluencedOrder, InfluencedRoad, CongestionLevel, TransformMatrix, CongestionFactor

Process: generate congestion level; repeat: generate congestion

**TOrderGenerator**

Attributes: DestinationLocation, InterActiveTime, TargetDepotID

Process: repeat: wait inter order time; generate order

**Conceptual model in PDL**

**TEventEffectGenerator**

Attributes: VenueLocation, EventStartTime, EventEndTime, InfluencedOrder, InfluencedRoad, CongestionLevel, TransformMatrix, CongestionFactor

VenueLocation: Set of coordinates

EventStartTime: Integer //predetermined

EventEndTime: Integer //predetermined

InfluencedOrder: Group of sets of coordinates for OrderGenerators which are influenced by the event

InfluencedRoad: Group of sequence numbers for Roads which are influenced by the event

EventScale: Integer //predetermined and divided into 5 ranks, rank 1 means Light, rank 2 means Moderate, rank 3 means Heavy, rank 4 means Severe, rank 5 means Gridlocked

TransformMatrix: Matrix of real //predetermined, shown in the Appendix

CongestionLevel: Integer, TableDistribution //discrete random distribution, determined by EventScale & TransformMatrix

CongestionFactor: Real //determined by CongestionLevel, illustrated in the Appendix

Process

Repeat

If Now == EventStartTime

newEventEffectGenerator = new TEventEffectGenerator //generate a new venue

newEventEffectGenerator.CongestionLevel = CongestionLevel.sample //table distribution

newEventEffectGenerator.CongestionFactor = CongestionFactor //determined by CongestionLevel

newEventEffectGenerator.Start

For TOrderGenerator.DestinationLocation InfluencedOrder

TOrderGenerator.InterActiveTime = TOrderGenerator.InterActiveTime. sample/ CongestionFactor //orders boom. If the interval from the last order generated is shorter than shortened interval time, new order generates now.

For TRoad.RoadID InfluencedOrder

TRoad.VelocityLimit = TRoad.VelocityLimit/CongestionFactor

//roads congest

If Now == EventEndTime

For TOrderGenerator.DestinationLocation InfluencedOrder

TOrderGenerator.InterActiveTime = TOrderGenerator.InterActiveTime. Sample \* CongestionFactor //orders recover

For TRoad.RoadID InfluencedOrder

TRoad.VelocityLimit = TRoad.VelocityLimit\*CongestionFactor

//roads recover

**TOrderGenerator**

DestinationLocation: Set of coordinates

InterActiveTime: ExponentialDistribution

TargetDepotID: Integer //the sequence number of the nearest depot

Process

Repeat

Wait InterActiveTime.sample

newOrder = new TOrder //Generate a new Order

newOrder.OrderTime = Now

newOrder.Destination = TOrderGenerator.DestinationLocation

newOrder.Volume = Volume.sample

newOrder.Start

Depot.Reactivate //activate the nearest depot

**TTruck**

Attributes: TruckLocation, OutDepotTime, ActiveTime, DeliveryArrivalTime, InDepotTime, DistanceEnergyConsumption, TimeEnergyConsumption, NrOrder, DepotBelonging

Process: wait for maintenance, deliver goods, plan VPR route, return to depot

TruckLocation: Set of coordinates //the mean-time location updated per unit time;

OutDepotTime: Integer //the time of a truck leaving the depot;

ActiveTime: Integer //the time of a truck being reactivated before the reparation;

DeliveryArrivalTime: Set of Integers //the set of time of a truck arriving each destination;

InDepotTime: Integer //the time of a truck back to the depot;

DistanceEnergyConsumption: Real //the energy consumption for a truck per unit distance in operation;

TimeEnergyConsumption: Real //the energy consumption for a truck per unit time (including the energy of driving and standby, such as air-conditioning and so on);

NrOrder: Integer //the number of unit space occupied on board

DepotBelonging: Integer //the sequence number of depot that a truck belongs to

Process

ActiveTime = Now

ServiceCentre = ServiceCentre

ServiceCentre.ServiceQueue.EnterQueue (self)

Wait ServiceCentre.ServiceTime

OutDepotTime = Now

DepotBelonging = null // delivery start

Get TRoad.VelocityLimit // retrieve the velocity on the road it’s on, for estimating travel time

Get Order.Destination & Order.OrderTime //get delivery destination, time window etc. Order-related information

Shortest route plan // use, e.g. dijkstra algorithm, to determine the route between depot to the closest order point.

While NrOrder >0

Move(1) //update for 1 simulation time, and make the vehicle advance

If TruckLocation == Order.Destination

NrOrder = NrOrder -Order.Volume

DeliveryArrivalTime = Now

Get all Depot.NrInStockTrucks // Retrieve the truck allocation in depots

Decide which depot to return // determine the the depot with least trucks to return

Shortest route plan

While TruckLocation != TDepot.DepotLocation

Move(1)

Depot.Reactivate

DepotBelonging = TDepot.DepotID

TDepot.NrInStockTrucks = TDepot.NrInStockTrucks+1 //Truck back to depot

Depot.Interrupt

Passivate

**TServiceCentre**

NrServiceCapacity: Integer

ServiceTime: ExponentialDistribution

ServiceQueue: Queue //queue for trucks waiting for service before delivery

Process

Repeat

While ServiceQueue is empty Standby

Truck = ServiceQueuee.FirstofQueue

Truck.LeaveQueue (ServiceQueue)

Wait ServiceTime.sample //generate distribution

**TDepot**

Attributes: DepotLocation, DepotID, NrInStockTrucks, OrderInStock, MaxOrder, MaxWaitingTime

Process: accept order, receive trucks

DepotLocation: Set of coordinates

DepotID: Integer //sequence number of depots

NrInStockTrucks: Integer //number of trucks belonging to the depot

OrderInStock: Integer //the number of units space that have not been distributed yet

MaxOrder: Integer //the max number of orders to deliver for a truck

Process

OrderInStock = OrderInStock+Order.Volume

If OrderInStock == MaxOrder //full-load case

NrInStockTrucks = NrInStockTrucks -1

Determine which truck to deliver

Truck.reactivate

NrInStockTrucks = 0

If Now-FirstOrder.OrderTime = MaxWaitingTime //not full-load case

NrInStockTrucks = NrInStockTrucks -1

Determine which truck to deliver

Truck.reactivate

NrInStockTrucks = 0

Passivate

**Initialization**

TTruck.TruckLocation = TDepot.DepotLocation

OrderGenerator.Start

EventEffectGenerator.Start

**Appendix**

EventScale: row

CongestionLevel: column

Table 1 The correspondence from CongestionLevel to CongestionFactor

|  |  |
| --- | --- |
| CongestionLevel | CongestionFactor |
| No congestion | 1 |
| 1 | 1.05 |
| 2 | 1.15 |
| 3 | 1.25 |
| 4 | 1.35 |
| 5 | 1.45 |