SUNIL CHOPRA

Technical Note: Routing and Scheduling in Transportation

The most important operational decision related to transportation in a supply chain is the routing and scheduling of deliveries. Managers must decide on the customers to be visited by a particular vehicle and the sequence in which they will be visited. For example, an online grocer such as Peapod is built on delivering customer orders to their homes. The success of its operations hinges on its ability to decrease transportation and delivery costs while providing the promised level of responsiveness to the customer. Given a set of customer orders, the goal is to route and schedule delivery vehicles such that the costs incurred to meet delivery promises are as low as possible. Typical objectives when routing and scheduling vehicles include a combination of minimizing cost by decreasing the number of vehicles needed, the total distance traveled by vehicles, and the total travel time of vehicles, as well as eliminating service failures such as a delays in shipments.

In this note, routing and scheduling problems are discussed from the point of view of the manager of a Peapod distribution center (DC). After customers place orders for groceries online, staff at the DC has to pick the items needed and load them on trucks for delivery. The manager must decide which trucks will deliver to which customers and the route that each truck will take when making deliveries. The manager must also ensure that no truck is overloaded and that promised delivery times are met.

One morning, the DC manager at Peapod has delivery orders from thirteen different customers. The DC's location, each customer on the grid, and the order size from each customer are shown in **Table 1**. The manager has four trucks, each capable of carrying up to 200 units. The manager believes that the delivery costs are strongly linked to the total distance the trucks travel, and that the distance between two points on the grid is correlated with the actual distance a vehicle will travel between those two points. The manager thus decides to assign customers to trucks and identify a route for each truck, with a goal of minimizing the total distance traveled.

Table 1: Customer Location and Demand for Peapod

	X-Coordinate	Y-Coordinate	Order Size a_i
Warehouse	0	0	
Customer 1	0	12	48
Customer 2	6	5	36
Customer 3	7	15	43
Customer 4	9	12	92
Customer 5	15	3	57
Customer 6	20	0	16
Customer 7	17	-2	56
Customer 8	7	-4	30
Customer 9	1	-6	57
Customer 10	15	-6	47
Customer 11	20	-7	91
Customer 12	7	-9	55
Customer 13	2	-15	38

The DC manager must first assign customers to be served by each vehicle and then decide on each vehicle's route. After the initial assignment, route sequencing and route improvement procedures are used to decide on the route for each vehicle. The DC manager decides to use the following computational procedures to support his decision:

- The savings matrix method
- The generalized assignment method

We discuss how each method can be used to solve the routing and scheduling decision at Peapod.

Savings Matrix Method

This method is simple to implement and can be used to assign customers to vehicles even when delivery time windows or other constraints exist. The major steps in the savings matrix method are:

1. Identify the distance matrix.

2. Identify the savings matrix.

3. Assign customers to vehicles or routes.

4. Sequence customers within routes.

The first three steps result in customers being assigned to vehicles, and the fourth step is used to route each vehicle to minimize the distance traveled.

Identify the Distance Matrix

The *distance matrix* identifies the distance between every pair of locations to be visited. The distance is used as a surrogate for the cost of traveling between the pair of locations. If the transportation costs between every pair of locations are known, the costs can be used in place of distances. The distance *Dist(A, B)* on a grid between a point *A* with coordinates (x_A, y_A) and a point *B* with coordinates (x_B, y_B) is evaluated as:

$$\text{Dist}(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}\,. \qquad (1)$$

The distance between every pair of locations for Peapod is shown in **Table 2**. The distances between every pair of locations are next used to evaluate the savings matrix.

Table 2: Distance Matrix for Peapod Deliveries

	DC	Cust 1	Cust 2	Cust 3	Cust 4	Cust 5	Cust 6	Cust 7	Cust 8	Cust 9	Cust 10	Cust 11	Cust 12	Cust 13
DC	0													
Cust 1	12	0												
Cust 2	8	9	0											
Cust 3	17	8	10	0										
Cust 4	15	9	8	4	0									
Cust 5	15	17	9	14	11	0								
Cust 6	20	23	15	20	16	6	0							
Cust 7	17	22	13	20	16	5	4	0						
Cust 8	8	17	9	19	16	11	14	10	0					
Cust 9	6	18	12	22	20	17	20	16	6	0				
Cust 10	16	23	14	22	19	9	8	4	8	14	0			
Cust 11	21	28	18	26	22	11	7	6	13	19	5	0		
Cust 12	11	22	14	24	21	14	16	12	5	7	9	13	0	
Cust 13	15	27	20	30	28	22	23	20	12	9	16	20	8	0

Identify the Savings Matrix

The *savings matrix* represents the savings that accrue on consolidating two customers on a single truck. Savings may be evaluated in terms of distance, time, or money. The manager at the Peapod DC constructs the savings matrix in terms of distance. A *trip* is identified as the sequence of locations a vehicle visits. The trip DC → Cust *x* → DC starts at the DC, visits customer *x,* and returns to the DC. The savings $S(x,y)$ is the distance saved if the trips DC → Cust *x* → DC and DC → Cust *y* → DC are combined to a single trip, DC → Cust *x* → Cust *y* → DC. These savings can be calculated by the following formula:

$$S(x,y) = Dist(\text{DC}, x) + Dist(\text{DC}, y) - Dist(x, y). \qquad (2)$$

For example, using Table 2 the manager evaluates $S(1,2) = 12 + 8 - 9 = 11$. The savings matrix for the Peapod deliveries is shown in **Table 3**. The savings matrix is then used to assign customers to vehicles or routes.

Table 3: Savings Matrix for Peapod Deliveries

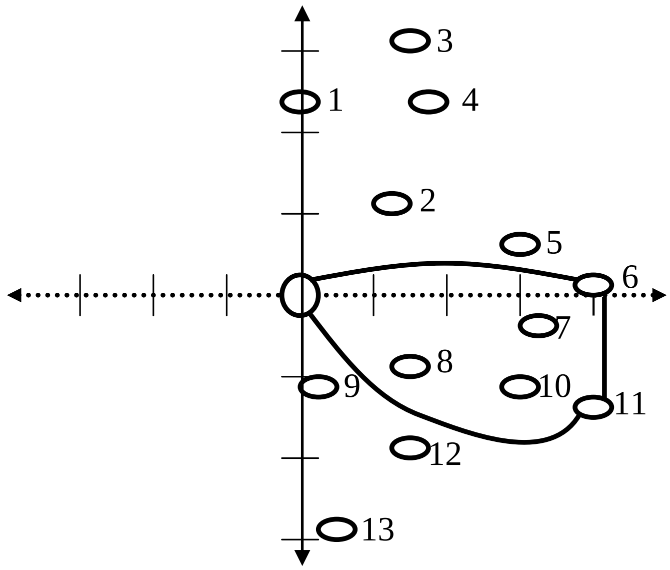
	Cust 1	Cust 2	Cust 3	Cust 4	Cust 5	Cust 6	Cust 7	Cust 8	Cust 9	Cust 10	Cust 11	Cust 12	Cust 13
Cust 1	0												
Cust 2	11	0											
Cust 3	21	15	0										
Cust 4	18	15	28	0									
Cust 5	10	14	18	19	0								
Cust 6	9	13	17	19	29	0							
Cust 7	7	12	14	16	27	33	0						
Cust 8	3	7	6	7	12	14	15	0					
Cust 9	0	2	1	1	4	6	7	8	0				
Cust 10	5	10	11	12	22	28	29	16	8	0			
Cust 11	5	11	12	14	25	34	32	16	8	32	0		
Cust 12	1	5	4	5	12	15	16	14	10	18	19	0	
Cust 13	0	3	2	2	8	12	12	11	12	15	16	18	0

Assign Customers to Vehicles or Routes

When assigning customers to vehicles, the manager attempts to maximize savings. An iterative procedure is used to make this assignment. Initially each customer is assigned to a separate route. Two routes can be combined into a *feasible* route if the total deliveries across both routes do not exceed the vehicle's capacity. At each iterative step, the Peapod manager attempts to combine routes with the highest savings into a new feasible route. The procedure is continued until no more combinations are feasible.

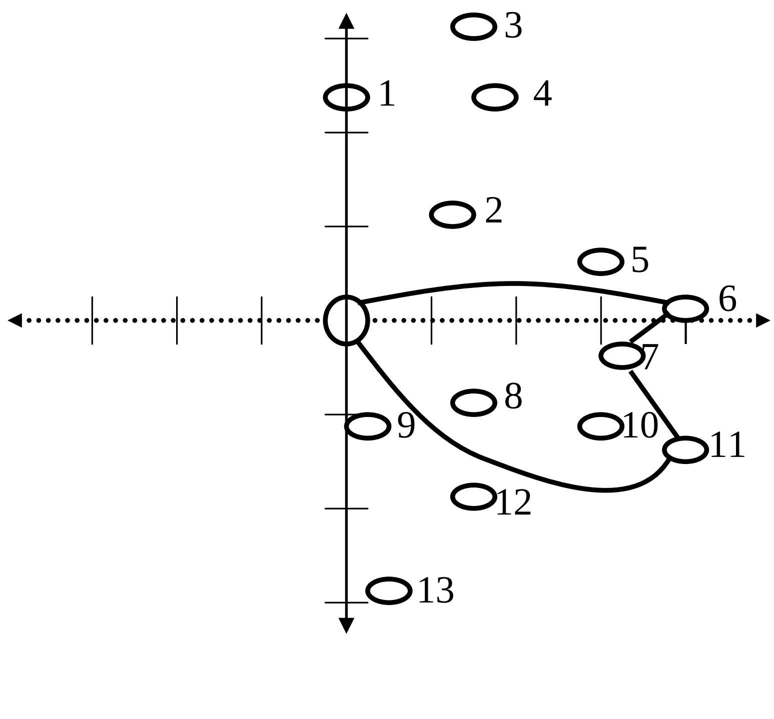
At the first step, the highest savings of 34 results on combining truck Routes 6 and 11. The combined route is feasible because the total load is 16 + 91 = 107, which is less than 200. The two customers are thus combined on a single route, as shown in **Figure 1**, and the saving of 34 is eliminated from further consideration.

Figure 1: Delivery Route by Assigning 6 and 11 to a Common Route



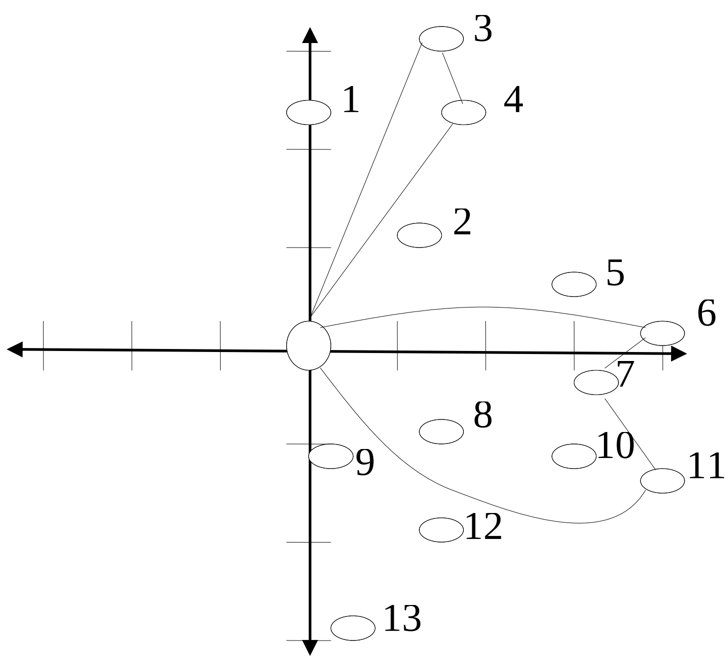
The next highest saving is 33, which results from adding Customer 7 to the route for Customer 6. This is feasible because the resulting load is 107 + 56 = 163, which is less than 200. Thus, Customer 7 is also added to Route 6, as shown in **Figure 2**.

Figure 2: Delivery Route by Assigning 6, 7, and 11 to a Common Route



The next highest saving now is 32, which results from adding Customer 10 to Route 6 (we need not consider the saving of 32 on combining Customer 7 with Customer 11 because both are already in Route 6). This, however, is not feasible, as Customer 10 has a delivery totaling 47 units and adding this amount to the deliveries already on Route 6 would exceed the vehicle capacity of 200. The next highest saving is 29, which results from adding either Customer 5 or 10 to Route 6. Each of these is also infeasible because of the capacity constraint. The next highest saving is 28, which results from combining Routes 3 and 4, which is feasible. The two routes are combined into a single route, as shown in **Figure 3**.

Figure 3: Delivery Route by Assigning 3 and 4 to a Common Route



Continuing the iterative procedure, the manager partitions customers into four groups {1, 3, 4}, {2, 9}, {6, 7, 8, 11}, {5, 10, 12, 13}, with each group assigned to a single vehicle. The next step is to identify the sequence in which each vehicle will visit customers.

Sequence Customers within Routes

At this stage the manager's goal is to sequence customer visits so as to minimize the distance each vehicle must travel. Changing the sequence in which deliveries are made can have a significant impact on the distance traveled by the vehicles. Consider the truck that has been assigned deliveries to Customers 5, 10, 12, and 13. If the deliveries are in the sequence 5, 10, 12, 13, the total distance traveled by the truck is 15 + 9 + 9 + 8 + 15 = 56 (distances are obtained from Table 2). In contrast, if deliveries are in the sequence 12, 5, 13, 10, the truck covers a larger distance of 11 + 14 + 22 + 16 + 16 = 79. Delivery sequences are determined by obtaining an initial route sequence and then using route improvement procedures to obtain delivery sequences with a lower transportation distance or cost.

ROUTE SEQUENCING PROCEDURES

The Peapod manager can use route sequencing procedures to obtain an initial trip for each vehicle. The initial trip is then improved using the route improvement procedure discussed later in this note. All route sequencing procedures are illustrated for the vehicle assigned to Customers 5, 10, 12, and 13.

Farthest Insert

Given a vehicle trip (including a trip consisting of only the DC) for each remaining customer, find the minimum increase in length for this customer to be inserted from all the potential points in the trip that they could be inserted. Then choose to actually insert the customer with the largest minimum increase to obtain a new trip. This step is referred to as a farthest insert because the customer farthest from the current trip is inserted. The process is continued until all remaining customers to be visited by the vehicle are included in a trip.

For the Peapod example, the manager is seeking a trip starting at the DC and visiting Customers 5, 10, 12, 13. The initial trip consists of just the DC with a length of 0. Including Customer 5 in the trip adds 30 to its length, including Customer 10 adds 32, including Customer 12 adds 22, and including Customer 13 adds 30 (see Table 2). Using the farthest insert, the manager adds Customer 10 to obtain a new trip (DC, 10, DC) of length 32.

At the next step, inserting Customer 5 in the trip raises the length of the trip to a minimum of 40, inserting Customer 12 raises it to 36, and inserting Customer 13 raises it to 47. The manager thus inserts the farthest Customer 13 to obtain the new trip (DC, 10, 13, DC) of length 47. This still leaves Customers 5 and 12 to be inserted. The minimum cost insertion for Customer 5 is (DC, 5, 10, 13, DC) for a length of 55, and the minimum cost insertion for Customer 12 is (DC, 10, 12, 13, DC) for a length of 48. The manager thus inserts Customer 5 to obtain a trip (DC, 5, 10, 13, DC) of length 55. Customer 12 is then inserted between Customers 10 and 13 to obtain a trip (DC, 5, 10, 12 13, DC) of length 56.

Nearest Insert

Given a vehicle trip (including a trip consisting of only the DC) for each remaining customer, find the minimum increase in length for this customer to be inserted from all the potential points in the trip that they could be inserted. Insert the customer with the smallest minimum increase to obtain a new trip. This step is referred to as a nearest insert because the customer closest to the current trip is inserted. The process is continued until all remaining customers the vehicle will visit are included in a trip.

For the Peapod example, the manager applies the nearest insert to the vehicle serving Customers 5, 10, 12, and 13. Starting at the DC, the nearest customer is 12. Inserting Customer 12 results in the trip (DC, 12, DC) of length 22. At the next step, inserting Customer 5 results in a trip of length 40, inserting Customer 10 results in a trip of length 36, and inserting Customer 13 results in a trip of length 34. Customer 13 results in the smallest increase and is inserted to obtain a trip (DC, 12, 13, DC) of length 34. The next nearest insertion is Customer 10 resulting in a trip (DC, 10, 12, 13, DC) of length 48, and the final insertion of Customer 5 results in a trip (DC, 5, 10, 12, 13, DC) of length 56.

Nearest Neighbor

Starting at the DC, this procedure adds the closest customer to extend the trip. At each step, the trip is built by adding the customer closest to the point last visited by the vehicle until all customers have been visited.

For the Peapod example, the customer closest to the DC is 12 (see Table 2). This results in the path (DC, 12). The customer closest to Customer 12 is 13, extending the path to (DC, 12, 13). The nearest neighbor of Customer 13 is 10 and the nearest neighbor of Customer 10 is 5. The Peapod manager thus obtains a trip (DC, 12, 13, 10, 5, DC) of length 59.

Sweep

In the sweep procedure, any point on the grid is selected (generally the DC itself) and a line is swept either clockwise or counterclockwise from that point. The trip is constructed by sequencing customers in the order they are encountered during the sweep.

The Peapod manager uses the sweep procedure with the line centered at the DC. Customers are encountered in the sequence 5, 10, 12, 13 to obtain the trip (DC, 5, 10, 12 13, DC) for a length of 56.

The initial trips resulting from each route sequencing procedure and their lengths are summarized in **Table 4**.

Table 4: Initial Trips Using Different Route Sequencing Procedures at Peapod

Route Sequencing Procedure	Resulting Trip	Trip Length
Farthest insert	DC, 5, 10, 12, 13, DC	56
Nearest insert	DC, 5, 10, 12, 13, DC	56
Nearest neighbor	DC, 12, 10, 5, 13, DC	59
Sweep	DC, 5, 10, 12, 13, DC	56

ROUTE IMPROVEMENT PROCEDURES

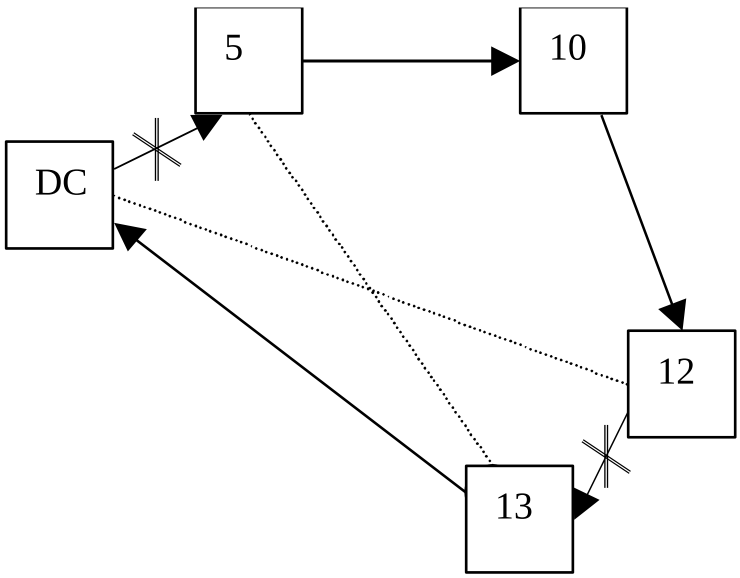
Route improvement procedures start with a trip obtained using a route sequencing procedure and improve the trip to shorten its length. The Peapod manager next applies route improvement procedures to alter the sequence of customers visited by a vehicle and to shorten the distance a vehicle must travel. The two route improvement procedures discussed are illustrated on the trip obtained as a result of the nearest neighbor procedure.

2-OPT

The 2-OPT procedure starts with a trip and breaks it at two places. This results in the trip breaking into two paths, which can be reconnected in two possible ways. The length for each reconnection is evaluated, and the smaller of the two is used to define a new trip. The procedure is continued on the new trip until no further improvement results.

For example, the trip (DC, 12, 10, 5, 13, DC) resulting from the nearest neighbor procedure can be broken into two paths (13, DC) and (12, 10, 5) and reconnected into the trip (DC, 5, 10, 12, 13, DC), as shown in **Figure 4**. The new trip has length 56, which is an improvement over the existing trip.

Figure 4: Improving Route Sequencing Using 2-OPT



3-OPT

The 3-OPT procedure breaks a trip at three points to obtain three paths that can be reconnected to form up to eight different trips. The length of each of the eight possible trips is evaluated and the shortest trip is retained. The procedure is continued on the new trip until no further improvement results.

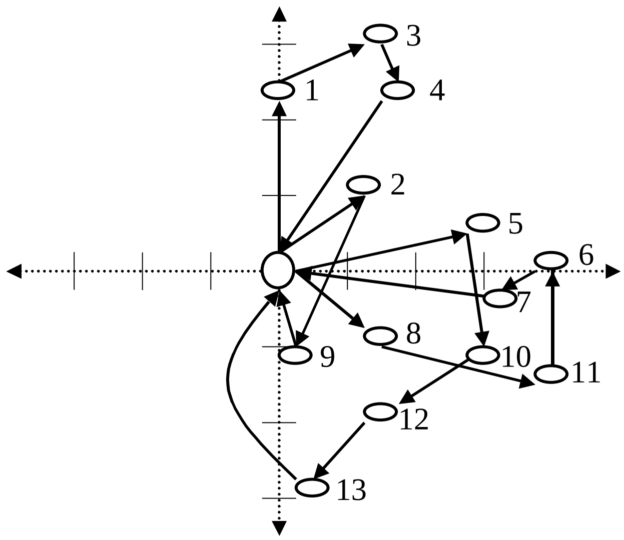
The trip (DC, 5, 10, 12, 13, DC) resulting from the 2-OPT procedure is broken up into three paths (DC), (5, 10), and (12, 13). The various resulting trips on reconnecting the three paths are (DC, 12, 13, 5, 10, DC) of length 65, (DC, 12, 13, 10, 5, DC) of length 81, and (DC, 13, 12, 5, 10, DC) of length 61. All other trips correspond to one of these four trips reversed. This application of the 3-OPT procedure does not improve the trip because the current trip is the shortest. At this stage the Peapod manager can form three new paths from the trip and repeat the procedure.

The Peapod manager uses route sequencing and improvement procedures to obtain delivery trips for each of the four trucks, as shown in **Table 5** and **Figure 5**. The total travel distance for the delivery schedule is 185.

Table 5: Peapod Delivery Schedule Using Saving Matrix Method

Truck	Trip	Length of Trip	Load on Truck
1	**DC, 2, 9, DC**	**32**	**93**
2	DC, 1, 3, 4, DC	39	183
3	DC, 8, 11, 6, 7, DC	58	193
4	DC, 5, 10, 12, 13, DC	56	197

Figure 5: Delivery Routes at Peapod Using Savings Matrix Method



Generalized Assignment Method

The generalized assignment method is more sophisticated than the savings matrix method and usually results in better solutions when there are few delivery constraints to be satisfied. The procedure for routing and sequencing of vehicles consists of the following steps:

1. Assign seed points for each route.

2. Evaluate insertion cost for each customer.

3. Assign customers to routes.

4. Sequence customers within routes.

The first three steps result in customers being assigned to a vehicle, and the fourth step identifies a route for each vehicle to minimize the distance traveled. We discuss each step in greater detail in the context of the delivery decision at Peapod.

Assign Seed Points for Each Route

The goal of this step is to determine a seed point corresponding to the center of the trip taken by each vehicle using the following procedure:

1. Divide the total load to be shipped to all customers by the number of trucks to obtain L_{seed}, the average load allocated to each seed point.

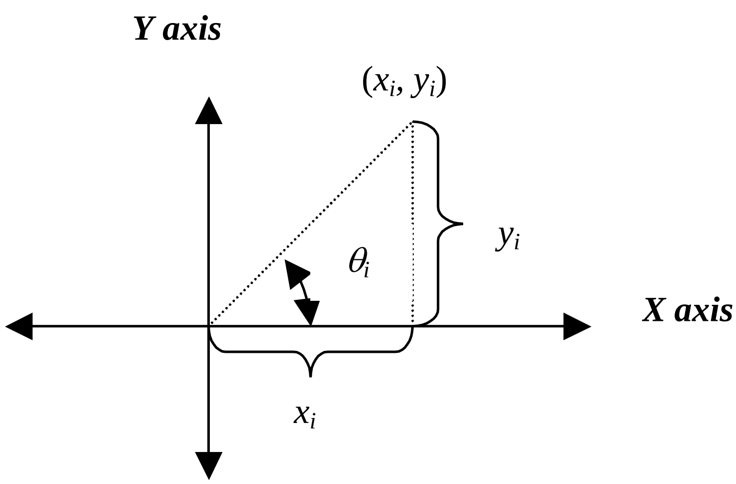
2. Starting at any customer, use a ray starting at the DC to sweep clockwise to obtain cones assigned to each seed point. Each cone is assigned a load of L_{seed}.

3. Within each cone, the seed point is located in the middle (in terms of angle) at a distance equal to that of the customer (with a partial or complete load allocated to the cone) farthest from the DC.

The manager at Peapod uses the procedure described earlier to obtain seed points for the deliveries described in Table 1. Given four vehicles and a total delivery load across all customers of 666 units, the manager obtains an average load per vehicle of $L_{seed} = 666 / 4 = 166.5$ units.

The next step is to sweep clockwise with a ray emanating from the DC to obtain four cones, one for each vehicle, including all customers. The first step in defining the cones is to obtain the angular position of each customer. The angular position (θ_i) of customer i with coordinates (x_i, y_i) is the angle made relative to the x axis by the line joining the customer i to the origin (DC), as shown in **Figure 6**.

Figure 6: Angular Position of Customer *i*



The angular position of each customer is obtained as the inverse tangent of the ratio of its y coordinate to the x coordinate.

$$\theta_i = \tan^{-1}(y_i/x_i). \quad (3)$$

The inverse tangent can be evaluated using the Excel function *ATAN()* as

$$\theta_i = \text{ATAN}(y_i/x_i). \quad (4)$$

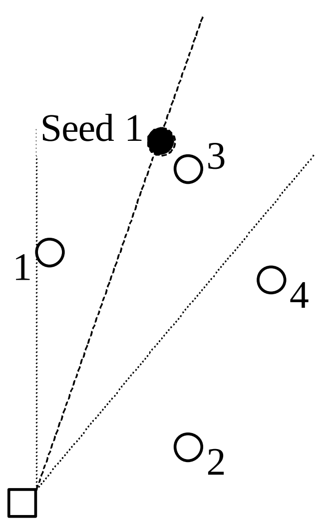
The angular position of each customer is obtained using Equation 4, as shown in **Table 6**.

Table 6: Angular Positions of Peapod Customers

	X Coordinate	Y Coordinate	Angular Position (Radians)	Demand
DC	0	0		
Customer 1	0	12	1.57	48
Customer 2	6	5	0.69	36
Customer 3	7	15	1.13	43
Customer 4	9	12	0.93	92
Customer 5	15	3	0.20	57
Customer 6	20	0	0.00	16
Customer 7	17	-2	-0.12	56
Customer 8	7	-4	-0.52	30
Customer 9	1	-6	-1.41	57
Customer 10	15	-6	-0.38	47
Customer 11	20	-7	-0.34	91
Customer 12	7	-9	-0.91	55
Customer 13	2	-15	-1.44	38

The next step is to sweep clockwise and order the customers as encountered. For Peapod, a clockwise sweep encounters customers in the order 1, 3, 4, 2, 5, 6, 7, 11, 10, 8, 12, and 9. Starting with Customer 1, four cones, each representing a load of L_{seed} = 166.5 units, are formed. Customers 1 and 3 combine to load 91 units on the truck. Customer 4 is encountered next in the sweep. Adding the entire load for Customer 4 would result in a load of 183, which is larger than L_{seed} = 166.5. To get a load of 166.5, only 166.5 – 91 = 75.5 units of the load should be included. Thus, the first cone extends to a point that is 75.5 / 92 of the angle between Customers 3 and 4. Customer 3 has an angular position of 1.13 and Customer 4 has an angular position of 0.93, resulting in an angle between them of 1.13 – 0.93 = 0.20. The first cone thus extends to an angle (75.5 / 92) × 0.20 beyond Customer 3 with a resulting angle of 1.13 – (75.5 / 92) × 0.20 = 0.97. The first cone thus has one end at Customer 1 (angle of 1.57) and the other at an angle of 0.97, as shown in **Figure 7**.

Figure 7: Sweep Method to Locate Seed 1



The seed point is then located at an angle α_1 = (0.97 + 1.57) / 2 = 1.27 in the middle of the cone at a distance equal to that of the farthest customer included. Customer 3, at a distance $d_1 = \sqrt{(7-0)^2 + (15-0)^2} = 17$, is the farthest customer in the first cone. Given the distance d_1, the coordinates (X_1, Y_1) of the Seed Point 1 are thus given by:

$X_1 = d_1 \cos(\alpha_1)$ = 17 cos(0.95) = 10, and $Y_1 = d_1 \sin(\alpha_1)$ = 17 sin(0.95) = 14.

The second cone starts at the angle 0.97 and includes 92 – 75.5 = 16.5 units of the Customer 4 load. On sweeping clockwise, Customers 2, 5, 6, and 7 are encountered before a load of 166.5 is exceeded. To get a load of exactly 166.5, only 41/56 of Customer 7 load is needed. The angular position of the end of the cone is thus 41/56 between Customers 6 and 7. Customer 6 is at angle of 0.00 and Customer 7 is at the angle of –0.12. The second cone thus ends at an angle of 0.00 – 0.12 × (41 / 56) = –0.09. The second cone has one end at an angle of 0.33 and the other at an angle of –0.09. The seed point is thus located at an angle α_2 in the middle of the cone; that is, α_2 = (0.33 – 0.09) / 2 = 0.12. The distance d_2 of the seed point for the second cone is the same as Customer 6, the farthest customer in the cone. This corresponds to a distance of $d_2 = 20$ (see Table 2). The coordinates (X_2, Y_2) of the Seed Point 2 are thus given by:

$X_2 = d_2 \cos(\alpha_2)$ = 20 cos(0.12) = 20, and $Y_2 = d_2 \sin(\alpha_2)$ = 20 sin(0.12) = 2.

Proceeding in the same manner, the Peapod manager forms four cones to determine the four seed points, as shown in **Table 8**.

Table 8: Seed Point Coordinates for Peapod Deliveries

Seed Point	X Coordinate	Y Coordinate
S_1	5	16
S_2	20	2
S_3	19	-5
S_4	5	-5

Evaluate Insertion Cost for Each Customer

For each Seed Point S_k and Customer i, the *insertion cost* c_{ik} is the extra distance that would be traveled if the customer is inserted into a trip from the DC to the seed point and back and is given by:

$c_{ik} = Dist(\text{DC}, i) + Dist(i, S_k) - Dist(\text{DC}, S_k)$,

where the *Dist()* function is evaluated as in Equation 1. For Customer 1 and Seed Point 1, the insertion cost is given by:

$c_{11} = Dist(\text{DC}, 1) + Dist(1, S_1) - Dist(\text{DC}, S_1) = 12 + 10 - 17 = 5$.

The Peapod manager evaluates all insertion costs c_{ik}, as shown in **Table 9**.

Table 9: Insertion Costs for Peapod Deliveries for Each Customer and Seed Point

Customer	Seed Point 1	Seed Point 2	Seed Point 3	Seed Point 4
1	2	14	18	23
2	2	2	5	11
3	2	15	21	30
4	4	10	15	25
5	15	0	4	21
6	25	2	5	29
7	22	2	1	22
8	11	2	0	3
9	12	7	4	3
10	24	5	0	19
11	32	10	4	29
12	20	8	4	8
13	30	20	15	18

Assign Customers to Routes

The manager next assigns customers to each of the four vehicles to minimize total insertion cost while respecting vehicle capacity constraints. The assignment problem is formulated as an integer program and requires the following input:

c_{ik} = insertion cost of Customer i and Seed Point k,
a_i = order size from Customer i,
b_k = capacity of Vehicle k.

Define the following decision variables:

y_{ik} = 1 if Customer i is assigned to Vehicle k, 0 otherwise.

The integer program for assigning customers to vehicles is given by:

$$Min \sum_{k=1}^{K} \sum_{i=1}^{n} c_{ik}\, y_{ik}$$

subject to:

$$\sum_{k=1}^{K} y_{ik} = 1,\ i = 1,\ldots,n,$$

$$\sum_{i=1}^{n} a_i\, y_{ik} \le b_k\,,\ k = 1,\ldots,K,$$

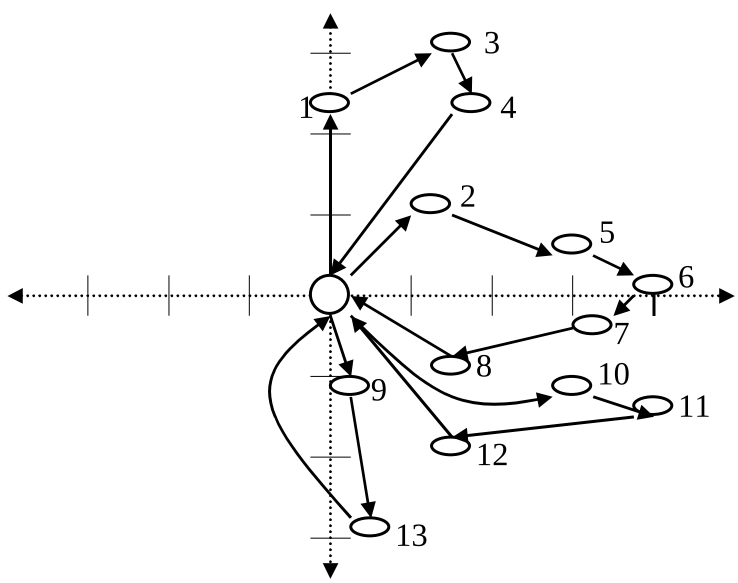
y_{ik} = 0 or 1, for all i and k.

For Peapod, the order size for each customer is given in Table 1, the insertion cost c_{ik} is obtained from Table 9, and the capacity of each vehicle is 200 units. The manager at Peapod solves the integer program using the Solver tool in Excel to obtain the assignment of customers to vehicles as shown in **Table 10** and **Figure 8**. The sequencing of customers within each trip is obtained using the route sequencing and route improvement procedures discussed earlier. The total distance traveled for the delivery schedule is 159.

Table 10: Peapod Delivery Schedule Using Generalized Assignment Method

Truck	Trip	Length of Trip	Load on Truck
1	DC, 1, 3, 4, DC	39	183
2	DC, 2, 5, 6, 7, 8, DC	45	195
3	DC, 10, 11, 12, DC	45	193
4	DC, 9, 13, DC	30	95

Figure 8: Delivery Routes at Peapod Using Generalized Assignment Method



Applicability of Routing and Scheduling Methods

The delivery schedule for Peapod resulting from the generalized assignment method in Table 10 is superior to the solution obtained from the savings matrix method in Table 5. The generalized assignment method is more sophisticated and generally gives a better solution than the savings matrix method when the delivery schedule has no constraints other than vehicle capacity. The main disadvantage of the generalized assignment method is that it has difficulty generating good delivery schedules as more constraints are included. For example, if Peapod has fixed time windows within which deliveries must be made to customers, it is difficult to use the generalized assignment method to generate a delivery schedule. The generalized assignment method is recommended if the constraints are limited to vehicle capacity or total travel time.

The main strength of the savings matrix method is its simplicity and robustness. The method is simple enough to be easily modified to include delivery time windows and other constraints and robust enough to give a reasonably good solution that can be implemented in practice. Its main weakness is the quality of the solution. It is often possible to find better delivery schedules using more sophisticated methods. The savings matrix method is recommended in case there are many constraints that need to be satisfied by the delivery schedule. Software packages for transportation planning and routing and scheduling of deliveries are available from many supply chain software companies.