Vehicle Routing

Field Service

When services require travel to the customer’s location, a method to develop vehicle routes quickly that minimizes time and distance traveled is an important consideration. An algorithm to perform this task will be developed and illustrated here.

On a typical Saturday, a college student may need to accomplish several tasks: work out at the gym, do some research in the library, go to the laundromat, and stop at a food market. Assuming no constraints on when these tasks may be done, the student faces no great obstacle in developing an itinerary that will require the least amount of time and distance traveled. The solution is straightforward and can be formulated in one’s head.

Many services likewise must develop itineraries, but in these cases, the solutions may not be as obvious as the college student’s. Examples range from Federal Express’s ground transportation pickup and delivery routes to bread deliveries at your local supermarket to a telephone repair-person’s route each day. Clearly, these cases require a useful tool to determine acceptable routing and scheduling without a great deal of hassle.

Enter G. Clarke and J. W. Wright, who developed the Clarke-Wright (C-W) algorithm to schedule vehicles operating from a central depot and serving several outlying points. In practice, the C-W algorithm is applied to a problem through a series of iterations until an acceptable solution is obtained. Practical applications of the algorithm may not necessarily be optimal, but the short amount of time and the ease with which it can be applied to problems that are not elementary and straightforward make it an extremely useful tool. The logic of this algorithm, which involves a savings concept, serves as the basis for the more sophisticated techniques available in many commercial software programs.

The C-W savings concept considers the savings that can be realized by linking pairs of “delivery” points in a system that is composed of a central depot serving the outlying sites. As a very simple example, consider Bridgette’s Bagel Bakery. Bridgette bakes her bagels during downtime at her brother Bernie’s Beaucoup Bistro. Then she must transport her bagels to two sidewalk concession stands that are run by her sisters, Bernadette and Louise. Each stand is located 5 miles from the Bistro, but the stands themselves are 6 miles apart. The layout may be represented graphically as follows:

```
  Concession 1
    △
     |
     |
  6 miles

     |
     |
  Concession 2
  5 miles

Bakery
  5 miles
```

In this situation, the C-W algorithm first looks at the cost of driving from the bakery to one concession and back to the bakery, and then driving to the second concession and back to the bakery. Therefore, the total cost is equal to the sum of the costs (in miles) of driving from 0 to 1 and returning ($2C_{01}$) and driving from 0 to 2 and returning ($2C_{02}$), or

\[
\text{Total cost} = 2C_{01} + 2C_{02}
\]
Bridgette's total cost for following this route is $2 \times 5$ (miles) + $2 \times 5$ (miles), or 20 miles.

The C-W algorithm next considers the savings that can be realized by driving from the bakery to one concession, then to the second concession, and finally back to the bakery. This route saves Bridgette the cost of one trip from concession 1 back to the bakery and of one trip from the bakery to concession 2, but it adds the cost of the trip from concession 1 to concession 2. The net savings $S_{ij}$ gained by linking any two locations $i$ and $j$ into the same route is expressed as

$$S_{ij} = C_{ii} + C_{jj} - C_{ij}$$

Bridgette would realize a net savings of 4 miles from linking the two concessions by creating one trip from the bakery to concession 1 and then traveling to concession 2 and returning to the bakery.

$$S_{12} = C_{01} + C_{02} - C_{12}$$
$$= 5 + 5 - 6 = 4$$

Admittedly, this example can be solved easily by inspection, and it does not require a sophisticated heuristic. Even so, it does serve as a convenient illustration of the savings concept that forms the basis of the C-W algorithm.

**The Clarke-Wright Algorithm**

Application of the C-W algorithm to a less obvious situation proceeds through five steps, which will be described as we put them to work helping Bridgette, who is expanding her bagel service to four concessions in outlying areas. The distances related to each of these concessions and the bakery are given in Exhibit 15.1.

![Network of Bakery and Four Concessions with Distances in Miles](image)

**EXHIBIT 15.1** Network of Bakery and Four Concessions with Distances in Miles

1. Construct a shortest-distance half-matrix (i.e., the matrix will contain the shortest distance between each pair of sites, including the starting location). A half-matrix is sufficient for this use because travel distance or time is the same in both directions. The shortest-distance half-matrix for Bridgette's bakery and the four outlying concessions is shown in Exhibit 15.2. (Note: For very large problems,
the shortest distances may not be obvious. In these cases, computer software to make these calculations is available.)

2. **Develop an initial allocation of one round-trip to each destination.** Note in the network beside Exhibit 15.2 that each concession is linked to the bakery by double lines with directional arrows. Four round-trips are represented.

<table>
<thead>
<tr>
<th>Concessions</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bakery</td>
<td>0</td>
<td>8</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>12</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>Concessions</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EXHIBIT 15.2  Shortest-Distance Half-Matrix:**
Miles between Bakery and Concessions

3. **Calculate the net savings for each pair of outlying locations, and enter them in a net savings half-matrix.** These net savings for each pair of outlying locations are calculated using the equation for \(S_{ij}\), just as we did in Bridgette’s initial problem. In this example, the net savings from linking concessions 1 and 2 is \(8 + 15 - 12 = 11\). Similar calculations are made for each of the other possible pairs, and the values then are entered into a net savings half-matrix, as shown in Exhibit 15.3.

<table>
<thead>
<tr>
<th>Concessions</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bakery</td>
<td></td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>11</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>Concessions</td>
<td>2</td>
<td>17</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EXHIBIT 15.3  Net Savings**
between All Concession Pairs

4. **Enter values for a special trip indicator** \(T\) **into appropriate cells of the net savings half-matrix.** Our net savings calculation for linking each pair is based on how much is saved relative to the cost of the vehicle making a round-trip to each member of the pair. We will add to our net savings half-matrix the indicator \(T\), which will show if two locations in question—for example, \(i\) and \(j\) or 0 (which represents the point of origin) and \(j\)—are directly linked. \(T\) may have one of three values, as given below:

a. \(T = 2\) when a vehicle travels from the point of origin (Bridgette’s bakery in our example) to location \(j\) (concession 1, 2, 3, or 4 in our example) and then returns. This is designated as \(T_{ij} = 2\) and will appear only in the first row of the half-matrix. The appropriate value of \(T\) is entered into the net savings half-matrix and circled to distinguish it from the savings value. Remember, \(T = 2\) indicates a round-trip.
b. \( T = 1 \) when a vehicle travels one way directly between two locations \( i \) and \( j \). This is designated as \( T_{ij} = 1 \) and can appear anywhere in the half-matrix. Remember, \( T = 1 \) indicates a one-way trip.

c. \( T = 0 \) when a vehicle does not travel directly between two particular locations \( i \) and \( j \). Accordingly, this is designated as \( T_{ij} = 0 \). Remember, \( T = 0 \) indicates that no trip is made between that pair of locations.

By convention, the \( T = 0 \) value is not entered; a cell without a \( T \) value of 1 or 2 noted in the matrix is understood to have a \( T = 0 \). It is important to recognize that for each location \( x \), the sum of the \( T \) values in column \( x \) plus the sum of the \( T \) values in row \( x \) must equal 2 (i.e., a vehicle must arrive and depart for every location served).

Exhibit 15.4a shows Bridgette’s net savings half-matrix for her four new concessions, with the appropriate \( T \) value of 2 listed in the cells representing round-trips between the bakery and each concession location. Note that the directional lines on the graphical depiction of this initial solution indicate a round-trip to each location.

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
2 & 2 & 2 & 2 \\
1 & 11 & 7 & 2 \\
2 & 17 & 14 & \\
3 & 9 & & \\
\end{array}
\]

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
0 & & & \\
\end{array}
\]

\begin{itemize}
  \item \textbf{Routes:} First trip: 0-1-0 \\
  \textbf{Second trip:} 0-2-0 \\
  \textbf{Third trip:} 0-3-0 \\
  \textbf{Fourth trip:} 0-4-0 \\
\end{itemize}

**EXHIBIT 15.4a  Initial Solution**

5. Identify the cell in the net savings half-matrix that contains the maximum net savings. If the maximum net savings occurs in cell \((i, j)\) in the half-matrix, then locations \( i \) and \( j \) can be linked if, and only if, the following conditions are met:

a. \( T_{0i} \) and \( T_{0j} \) must be greater than zero.

b. Locations \( i \) and \( j \) are not already on the same route or loop.

c. Linking locations \( i \) and \( j \) does not violate any system constraints, which will be discussed later.

If all three conditions are met, set \( T_{ij} = 1 \). In Bridgette’s case, cell \((2, 3)\) has the highest net savings (i.e., 17). \( T_{02} \) and \( T_{03} \) are each greater than zero, locations 2 and 3 are not already on the same route, and at present, there are no constraints to linking locations 2 and 3. Thus, all conditions are met, and we may enter a \( T \) value of 1 in cell \((2, 3)\), as shown in Exhibit 15.4b. This \( T_{23} = 1 \) in the cell indicates a one-way trip between concessions 2 and 3. At the same time that we have established the one-way trip between locations 2 and 3, we have eliminated a one-way trip from location 2 back to the bakery (0) and another one-way trip from the bakery to location 3. Therefore, it is necessary to reduce the \( T = 2 \) values in cells \((0, 2)\) and \((0, 3)\) to \( T = 1 \) in each. Exhibit 15.4b shows the appropriate \( T \) values for this new iteration, and the graphical depiction indicates the three new one-way routes.
EXHIBIT 15.4b  First Iteration

If any one of the conditions—5a, 5b, or 5c—is not met, then identify the cell with the next highest savings and repeat step 5. If necessary, repeat this inspection until you have identified the cell with the highest savings that satisfies all three conditions, and set its $T$ value equal to 1 (remember to reduce the appropriate $T = 2$ or $T = 1$ values in row 0). If no cell meets the conditions, then the algorithm ends. (The algorithm also ends when all locations are linked together on a single route, which we will discover as we proceed with Bridgette’s problem.)

This first application of the C-W algorithm has saved Bridgette 17 miles, but still more savings can be realized by subjecting her data to another iteration of step 5. Looking again at Exhibit 15.4b, we can identify cell (2, 4) as having the next highest net savings value (i.e., 14). $T_{02}$ and $T_{04}$ are each greater than zero, locations 2 and 4 are not on the same route at present, and there are no constraints against having locations 2 and 4 on the same route. Therefore, we can link these two locations. Enter $T = 1$ in cell (2, 4), and reduce each of the $T$ values in cells (0, 2) and (0, 4) by one trip, as shown in Exhibit 15.4c. Note in the graphical depiction that the trips from the bakery to concession 2 and from concession 4 back to the bakery have been eliminated, thus requiring an adjustment of the directional arrows.

EXHIBIT 15.4c  Second Iteration

Is further improvement possible? The next highest net savings is 11, found in cell (1, 2). In this situation, $T_{01}$ is greater than zero, but $T_{02}$ is not. Therefore, linking these two locations would violate condition 5a. Moving on, cell (3, 4) has the next highest net savings. The $T_{03}$ and $T_{04}$ values are each greater than zero, but concessions 3 and 4 are already on the same route, which violates condition 5b. Therefore, we must look at the next highest net savings, which is 7, in cell (1, 3). Here, $T_{01}$ and $T_{03}$ are each greater than zero, concessions 1 and 3 are not already on the same route, and no constraints exist. Therefore, we may link these two locations. We enter $T = 1$ in cell (1, 3) and reduce the $T$ values in cells (0, 1) and (0, 3) by 1 each, as shown in Exhibit 15.4d.
We have removed one trip between the bakery and concession 1 and one trip from concession 3 to the bakery. The directional arrows suggest that a counterclockwise route be used, but our assumption of equal time or distance traveling in either direction would permit the final route to be traversed in either direction.

EXHIBIT 15.4d  Final Solution

Vehicle Routing with Constraints

Suppose that business booms and Bridgette decides to supply four new franchise operations, and that these franchises are located according to the schematic shown in Exhibit 15.5. Unfortunately, Bridgette cannot carry enough bagels in her Blue Bagel Beamer to supply all the new locations on a single route such as the one we constructed in the previous section. Each franchise requires 500 bagels per day, and she can transport a maximum of 1,000 bagels per trip. How can we use the C-W algorithm to solve Bridgette’s problem? In general, the introduction of a constraint such as Bridgette’s capacity limit or a delivery time window does not alter the method of applying the algorithm. We need only account for the constraint so it does not violate step 5c.

EXHIBIT 15.5  Network Representation of a Single Bakery and Four Concessions

In the present example, our first step again is to construct a shortest-distance half-matrix containing the distance between each pair of locations, as shown in Exhibit 15.6.