

Clarke & Wright's Savings Algorithm

Jens Lysgaard

(translated by Michael M. Sørensen)

Department of Management Science and Logistics

The Aarhus School of Business

Fuglesangs Allé 4

DK-8210 Aarhus V

September 1997

1. Introduction.

In 1964 Clarke & Wright published an algorithm for the solution of that kind of vehicle routing problem, which is often called the classical vehicle routing problem. This algorithm is based on a so-called savings concept.

This note briefly describes the algorithm and demonstrates its use by an example.

2. Problem characteristics.

The vehicle routing problem, for which the algorithm has been designed, is characterized as follows. From a depot goods must be delivered in given quantities to given customers. For the transportation of the goods a number of vehicles are available, each with a certain capacity with regard to the quantities. Every vehicle that is applied in the solution must cover a route, starting and ending at the depot, on which goods are delivered to one or more customers.

The problem is to determine the allocation of the customers among routes, the sequence in which the customers shall be visited on a route, and which vehicle that shall cover a route.

The objective is to find a solution which minimizes the total transportation costs. Furthermore, the solution must satisfy the restrictions that every customer is visited exactly once, where the demanded quantities are delivered, and the total demand on every route must be within the vehicle's capacity.

The transportation costs are specified as the cost of driving from any point to any other point. The costs are not necessarily identical in the two directions between two given points.

3. The savings algorithm.

The savings algorithm is a heuristic algorithm, and therefore it does not provide an optimal solution to the problem with certainty. The method does, however, often yield a relatively good solution. That is, a solution which deviates little from the optimal solution.

The basic savings concept expresses the cost savings obtained by joining two routes into one route as illustrated in figure 1, where point 0 represents the depot.

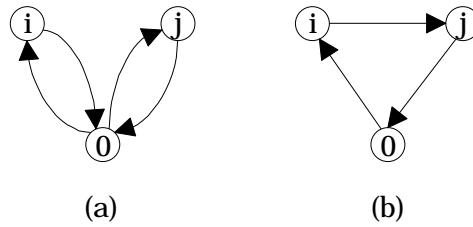


Figure 1. Illustration of the savings concept.

Initially in figure 1(a) customers *i* and *j* are visited on separate routes. An alternative to this is to visit the two customers on the same route, for example in the sequence *i-j* as illustrated in figure 1(b). Because the transportation costs are given, the savings that result from driving the route in figure 1(b) instead of the two routes in figure 1(a) can be calculated. Denoting the transportation cost between two given points *i* and *j* by c_{ij} , the total transportation cost D_a in figure 1(a) is:

$$D_a = c_{0i} + c_{i0} + c_{0j} + c_{j0}$$

Equivalently, the transportation cost D_b in figure 1(b) is:

$$D_b = c_{0i} + c_{ij} + c_{j0}$$

By combining the two routes one obtains the savings S_{ij} :

$$S_{ij} = D_a - D_b = c_{i0} + c_{0j} - c_{ij}$$

Relatively large values of S_{ij} indicate that it is attractive, with regard to costs, to visit points i and j on the same route such that point j is visited immediately after point i .

There are two versions of the savings algorithm, a sequential and a parallel version. In the sequential version exactly one route is built at a time (excl. routes with only one customer), while in the parallel version more than one route may be built at a time.

In the first step of the savings algorithm the savings for all pairs of customers are calculated, and all pairs of customer points are sorted in descending order of the savings. Subsequently, from the top of the sorted list of point pairs one pair of points is considered at a time. When a pair of points i - j is considered, the two routes that visit i and j are combined (such that j is visited immediately after i on the resulting route), if this can be done without deleting a previously established direct connection between two customer points, and if the total demand on the resulting route does not exceed the vehicle capacity. In the sequential version one must start anew from the top of the list every time a connection is established between a pair of points (since combinations that were not viable so far now may have become viable), while the parallel version only requires one pass through the list.

The actual way the algorithm works is illustrated in the following by means of a numerical example.

We consider a problem with 5 customers. The transportation costs between all pairs of points are shown in the following table, where 0 represents the depot (the costs are symmetric, and for that reason only the upper half of the table is filled in).

		To					
		0	1	2	3	4	5
From	0	-	28	31	20	25	34
	1		-	21	29	26	20
	2			-	38	20	32
	3				-	30	27
	4					-	25
	5						-

The customers' demands that must be delivered from the depot are given in the following table. The vehicle capacity is 100 units.

Customer	Quantity
1	37
2	35
3	30
4	25
5	32

The savings S_{ij} are calculated to the following values (only the upper half of the table is shown, since the savings are symmetric due to symmetric costs):

		j				
		1	2	3	4	5
i	1	-	38	19	27	42
	2		-	13	36	33
	3			-	15	27
	4				-	34
	5					-

Now the point pairs are sorted in descending order of the savings. This gives the following sorted list of point pairs:

1-5

1-2

2-4

4-5

2-5

1-4

3-5

1-3

3-4

2-3

The sequential savings algorithm

In the example customers 1 og 5 are considered first. They can be assigned to the same route since their joint demand for 69 units does not exceed the vehicle capacity. Now we establish the connection 1-5, and thereby points 1 and 5 will be neighbors on a route in the final solution.

Next we consider customers 1 and 2. If customers 1 and 2 should be neighbors on a route, this would require the customer sequence 2-1-5 (or 5-1-2) on a route, because we have established already that 1 and 5 must be visited in immediate succession on the same route. The total demand (104) on this route would exceed the vehicle capacity (100). Therefore, customers 1 and 2 are not connected.

If points 2 and 4, which is the next pair in the list, were connected at this stage, we would be building more than one route (1-5 and 2-4). Since the sequential version of the algorithm is limited to making only one route at a time, we disregard the point pair 2 and 4.

The combination of the next pair of points, 4 and 5, results in the route 1-5-4 with a total demand of 94. This combination is feasible, and we establish the connection between 4 and 5 as a part of the solution. Running through the list we find that due to the capacity restriction no more points can be added to the route. Thereby we have formed the route 0-1-5-4-0. In the next pass of the savings list we only find the point pair 2 and 3. These two points can be visited on the same route, and we make the route 0-2-3-0.

The sequential algorithm has constructed a solution with two routes. The transportation costs for the route 0-1-5-4-0 are 98, and for the route 0-2-3-0 the transportation costs are 89. The total transportation costs are, therefore, 187.

The parallel savings algorithm

In the parallel version 1 and 5 are also combined first. Since the parallel algorithm may build more than one route at a time, points 2 and 4 are also combined. Finally, points 3 and 5 are combined. In this way the algorithm constructs the routes 0-1-5-3-0 and 0-2-4-0 with total transportation costs amounting to 171.

It is worth noting that the number of routes may be reduced during the process of the parallel algorithm. For example, the two routes 0-1-2-0 and 0-3-4-0 will be combined into one route if the connection from 2 to 3 is established; in that case the resulting route becomes 0-1-2-3-4-0.

As it also turns out in the present example, the parallel savings algorithm frequently provides better results than the sequential algorithm. Dependent upon the way the algorithms are implemented, the parallel algorithm may also involve more computational work in connection with the management of several routes at the same time. Therefore, it cannot be stated in general whether the sequential or the parallel algorithm is more appropriate.

Literature

Clarke, G. & Wright, J.W.: "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, Vol. 12, 1964, pp. 568-581.