



## The Problem of Optimal Delivery of Frozen and Chilled Goods with Given Priority from Multiple Warehouses

---

Jaromír Zahrádka

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 25, 2024

# The Problem of Optimal Delivery of Frozen and Chilled Goods with Given Priority from Multiple Warehouses

Jaromír Zahrádka<sup>1</sup>

**Abstract.** This article comes up with a specific example of the vehicle routing problem. The selling company has to deliver the ordered frozen or chilled goods from  $m$  warehouses to  $n$  customers as efficiently as possible. Each customer has ordered goods stored in a certain number of containers which need to be transported. All customer points of delivery and warehouse points are given by GPS coordinates. The objective of the solution is to select the number of vehicles and their routes between suitable warehouse and customers in such a way that the total travel distance or travel time is as short as possible. The order of customers on each route respects the priority of delivery of frozen goods over chilled ones. This means that chilled goods are unloaded from the truck only after all frozen goods have been unloaded at previous customer delivery points. Each of these delivery points is visited only once by one of the vehicles. In each warehouse, the same number of trucks ends the journey as they left. All trucks have the same pre-limited capacity of containers. In this article, the algorithm of the vehicle routing problem with multiple warehouses and priority of delivery of frozen goods was created and implemented in Matlab code.

**Keywords:** Chilled goods, frozen goods, Matlab, mixed integer linear programming, optimization, point of delivery, priority, vehicle routing problem

**JEL Classification:** C64

**AMS Classification:** 68W04, 90C11, 05C20

## 1 The Vehicle Routing Problem with Multiple Warehouses.

The vehicle routing problem with multiple warehouses (VRPMW) is inspired by a practical situation that a trading company supplying, for example, food products has several warehouses in its territory from which it distributes goods to customers. The entire range of delivered goods is available in all warehouses and there is always a sufficient number of vehicles available. The solution to the VRPMW consists in planning the routes of trucks delivering goods so that the total distance travelled by all vehicles is as small as possible.

Our presented multi-storage VRPMW solution with customer priority according to the temperature regime of the delivered goods is a continuation of the single-storage VRP solution [1], and the Matlab one published in [6]. Our solution uses the principles of integer programming, which are contained in [4, 5]. They respect the principles of operating intelligent transport systems [2].

### 1.1 Mathematical Formulation of VRPMW

We calculate that  $n$  customers are served from  $s$  warehouses. The customers sites (nodes) are marked by numbers  $1, \dots, n$ , and warehouses (depots) are marked by numbers  $n + 1, \dots, n + s$ . The vehicle routing problem with multiple warehouses (VRPMW) can be presented as the subsequent graph problem. Let  $G = (V, E)$  be a complete directed graph where  $V = \{1, \dots, n, n + 1, \dots, n + s\}$  is the nodes set and  $E$  is the set of all oriented arcs. Nodes  $i = 1, 2, \dots, n$  correspond to the customers, each with a number  $q_i$  of demand containers, which form the row vector  $\mathbf{q} = (q_1, q_2, \dots, q_n)$ . Each oriented  $arc(i, j)$  is associated with non-negative values  $d_{ij}$  of travel distance (in meters), and  $c_{ij}$  of travel time (in sec) from node  $i$  to node  $j$ . For easier reference we assume,  $I = \{1, \dots, n\}$ , and  $I_D = I \cup \{n + 1, \dots, n + s\}$ . The distance matrix  $\mathbf{D} = (d_{ij})_{i, j \in I_D}$  and time distance matrix  $\mathbf{C} = (c_{ij})_{i, j \in I_D}$  are non-negative and generally asymmetric.

The VRPMW consists of finding a collection of simple cycles and simple paths between warehouses, each corresponding to a vehicle route with minimal sum of the distances of the cycles and path arcs, such that:

- each cycle starts and ends in one of the warehouses;
- each path starts in one of the warehouses and ends in another one;
- each vertex  $j \in I$  is visited by exactly one cycle or path;

---

<sup>1</sup> University of Pardubice, Department Mathematics and Physics, Studentská 95, 53210 Pardubice, jaromir.zahradka@upce.cz.

- d) the same number of cycles and paths start and end in each warehouse;
- e) the sum of delivered containers during a cycle or path not exceed the vehicle capacity  $Q$ .

At the beginning and end of each cycle or path is one of  $s$  stores. All customers who have ordered frozen goods must be visited. After the delivery of all frozen goods, the customers to whom the chilled goods are delivered are visited.

We assume, that  $m_i$  is the service time associated with the unloading of goods and dealing with the customer for each customer  $i \in I$ . All service times are arranged in a row vector  $\mathbf{m} = (m_1, m_2, \dots, m_n)$ .

## 1.2 Mathematical Solution of VRPMW

The used optimization model is generally described by

$$\min_{\mathbf{V}} (\mathbf{f} \cdot \mathbf{V}) \text{ subject to } \begin{cases} \mathbf{V}_{intcon} \text{ are integers} \\ \mathbf{A} \cdot \mathbf{V} \leq \mathbf{b} \\ \mathbf{A}_{eq} \cdot \mathbf{V} = \mathbf{b}_{eq} \\ \mathbf{I}_b \leq \mathbf{V} \leq \mathbf{u}_b \end{cases} \quad (1)$$

The vector  $\mathbf{V}$  is the column vector of all flow variables;  $\mathbf{f} \cdot \mathbf{V}$  is the objective function, the coefficients of this objective function are components of the row vector  $\mathbf{f}$ ;  $\mathbf{V}_{intcon}$  is the list of variable indices of the vector  $\mathbf{V}$  that takes only the integer values;  $\mathbf{A} \cdot \mathbf{V} \leq \mathbf{b}$  determines the system of inequality constraints;  $\mathbf{A}_{eq} \cdot \mathbf{V} = \mathbf{b}_{eq}$  determines the system of linear equations; and inequalities  $\mathbf{I}_b \leq \mathbf{V} \leq \mathbf{u}_b$  indicate the lower and upper bounds of flow variables.

The core of the solution of VRPMW is finding suitable configurations of paths and cycles in the graph that start and end in one of the warehouses. Each customer is on exactly one path or cycle. Another condition is that the same number of vehicles arrive at each warehouse, as left the warehouse. The optimal solution is the configuration of paths and cycles that gives the shortest total sum of the lengths of all paths and cycles. To realize the solution by the optimization method, integer flow variables  $x_{ij}$  are introduced on the set of all oriented arcs of the complete oriented graph, which can only take on binary values 0 or 1. The value  $x_{ij} = 1$  means that the arc from  $i$  node to  $j$  is included in one path or cycle, and the value  $x_{ij} = 0$  means that the corresponding arc is not included. For systemic reasons, variables  $x_{ii}$  are used but are fixed  $x_{ii} = 0$ , for each  $i \in I_D$ . Variables  $x_{ij}$  are elements of a matrix  $\mathbf{X} = (x_{ij})_{ij \in I_D}$ , and the number of  $x_{ij}$  variables is  $(n + s)^2$ .

In our work we use other specific integer flow variables,  $y_i$ , for each  $i \in I$ , which indicate the number of containers that were unloaded to customers during the journey from the depot up to and including the node  $i$ . The variables  $y_i$  are  $n$  elements of the vector  $\mathbf{y} = (y_1, y_2, \dots, y_n)$ . The sum of all flow variables is  $(n + m)^2 + n$ .

Elements of matrix  $\mathbf{X} = (x_{ij})_{ij \in I_D}$  and vector  $\mathbf{y} = (y_1, y_2, \dots, y_n)$  are arranged in the column vector  $\mathbf{V}$  so that,  $\mathbf{V}^T = (x_{1,1}, x_{1,2}, \dots, x_{1,n+m}, x_{2,1}, x_{2,2}, \dots, x_{1,n+m}, \dots, x_{n+m,1}, x_{n+m,2}, \dots, x_{n+m,n+m}, y_1, y_2, \dots, y_n)$ .

The solution of VRPMW is the optimal solution of a mixed-integer linear programming problem:

$$\min_{(\mathbf{X}, \mathbf{y})} \{ \sum_{i,j=0}^{n+s} d_{ij} \cdot x_{ij} + \sum_{i=1}^n y_i \} \text{ subject to} \quad (2)$$

$$x_{ij}, \quad i, j \in I_D \text{ are binary, i.e. } x_{ij} \in \{0,1\} \quad (3)$$

$$y_i, \quad i \in I \text{ are integer} \quad (4)$$

$$Q x_{ij} + y_i - y_j \leq Q - q_j, \quad i, j \in I, \quad i \neq j \quad (5)$$

$$Q x_{kj} - y_j \leq Q - q_j, \quad k \in I_D - I, j \in I \quad (6)$$

$$\sum_{j \in I_D} x_{ij} = 1, \quad i \in I \quad (7)$$

$$\sum_{i \in I_D} x_{ij} = 1, \quad j \in I \quad (8)$$

$$\sum_{j \in I} x_{kj} - \sum_{i \in I} x_{ik} = 0, \quad k \in I_D - I \quad (9)$$

$$x_{ii} = 0, \quad i \in I_D \quad (10)$$

$$x_{ij} = 0, \quad i, j \in I_D - I, \quad (11)$$

$$0 \leq x_{ij} \leq 1, \quad i, j \in I_D \quad (12)$$

$$q_j \leq y_j \leq Q, \quad j \in I \quad (13)$$

It means, that the linear optimization function is minimized such that

$$f = \sum_{i,j=1}^{n+s} d_{ij}x_{ij} + \sum_{i=0}^n y_i. \quad (14)$$

The sum of distances (in meters)  $\sum_{i,j=1}^{n+s} d_{ij}x_{ij}$  guarantees finding a collection of cycles and paths such that the sum of their lengths is minimal. The part  $\sum_{i=1}^n y_i$  of the optimization function (13) is about four orders of magnitude smaller and does not affect the optimization according to the smallest length. This part guarantees that the solutions of flow values  $y_i$  are also as small as possible. Constraint (5) defines  $n(n-1)$  conditions between flow variables  $y_i, y_j$  and  $x_{ij}$ . In the case  $x_{ij} = 1$ , the inequality (5) expresses the relationship  $y_j \geq y_i + q_j$ . Thanks to the inclusion of the variables  $y_1, y_2, \dots, y_n$  in the optimized function (13), it is ensured that the values of  $y_j$  will always be minimal, i.e. the equation  $y_j = y_i + q_j$  will be applied instead of an inequality. In the case  $x_{ij} = 0$ , for  $i \neq j$ , the inequality (5) expresses the relationship  $y_j \geq y_i + q_j - Q$ .

Statements (7) and (8) declare  $2n$  equations, which express that only one arc leads from each node  $i \in I$  and only one arc leads to each node  $j \in I$ . Statement (9) declares that for each store nodes  $n+1, \dots, n+s$ , the number of arcs leaving is equal to the number of arcs entering. Statement (10) declares that there are no loops in the graph. Statement (11) declares that there are no arcs between store nodes.

The inequalities in (12) state that the lower and upper bounds of flow variables  $x_{ij}$  are 0 and 1. The inequalities in (13) declare that each flow variable  $y_j$  is greater than or equal to  $q_j$ , and each  $y_j$  can't be greater than the capacity  $Q$  of vehicle.

### 1.3 Mathematical Solution of VRPMW with Given Priority of Delivery

For the delivery of goods in two different temperature regimes (e.g. frozen goods  $-18^\circ\text{C}$  and chilled goods  $+6^\circ\text{C}$ ), they can be transported with a separating partition: The partition enables the respective temperature regimes to be maintained in individual parts of the transport space. In such a case, the sequence of customers should be prepared so that frozen goods are unloaded first, followed by customers with chilled goods. It means that customers (nodes) with frozen goods have a higher priority delivery.

A row vector  $p = (p_1, p_2, \dots, p_n)$ . with component values of 1 or 2 is introduced to indicate the priority of customer nodes. The node priority condition does not allow a cycle or path with any arc leading from a node with priority 2 to a node with priority 1. This is achieved by adding a new constraint

$$x_{ij} = 0, \quad i, j \in I, \quad p_i > p_j \quad (15)$$

in the optimization model (2).

### 1.4 Implementation of the Solution in Matlab Code

The solution to the VRPMW problem is implemented in the Matlab programming language and is contained in the M-function *SOLVER\_VRPMW\_Z.m*, which is presented as an Appendix at the end of this article. In the Matlab solution, the same variables are used as in the mathematical description. However, variable identifiers are adapted to the syntax of the Matlab language.

The M-function *SOLVER\_VRPMW\_Za.m* is started using the special setup M-script, which is not published in this article. Using the setup M-script, all input data are introduced, all output data are taken, and the all figures shown in this article are plotted.

The output flow variables are the components of column vector  $V$ , which is obtained as the result of the command  $V = \text{intlinprog}(f, \text{intcon}, A, b, Aeq, beq, lb, ub, [], options)$  (App. row No. 23). A more detailed explanation of command *intlinprog* can be found in the User's Guide [3].

The matrix  $A$  of system linear inequalities and the column vector  $b$  of their right sides are created in Matlab code by statements on lines No. 4 to 10 in the Appendix. The matrix  $Aeq$  of system linear equalities and the column vector  $beq$  of their right sides are created by statements on lines No. 11 to 15 of the Appendix.

The lower and upper bounds of the flow variables with respect to the relations (10), (11), (12), (13) are created as components of vectors  $Lb$  and  $ub$  on lines No. 16, 17, 18, 19 in the Appendix. The fulfilment of condition (15) is ensured by the command in line No. 20.

The coefficients of the objective function are created by commands on line No. 22 of the Appendix. By last command on the line No. 22 all flow variables are taken as integers.

The for-cycle commands on lines No. 26 and 27 allow calculation of the total length of cycles and paths  $TotLgth$  and the total driving time  $TotDur$  of all used trucks. The total driving time  $TotDur$  includes service times of all customers. The M-function  $VRPMW\_SOLVER\_Z.m$  is executed by using a special startup M-script which enables the introduction of input data, processing of output data, and drawing output cycles and paths. The startup M-script is not presented in this article.

## 2 Demonstration Applied Task

Three warehouses in the region of Czech Republic and 27 customers were selected for the demonstration of the applied task. The GPS coordinates of customer locations and warehouses are listed in Table 1. In the table  $q_i$  means the number of ordered containers by customer  $i$  and the value  $p_i$  specifies frozen or chilled goods. The value  $p_i = 1$  is for frozen goods and  $p_i = 2$  means chilled ones. The customer and warehouse locations according to their GPS coordinates you can see in Figure 1. The considered cargo capacity of all vehicles is 18 containers. The service time  $m_i$  for each customer  $i$  is given by 20 minutes plus one minute for each unloaded container  $q_i$ , i.e.  $m_i = (20 + q_i) 60$  in seconds.

$i$	Customers									
	1	2	3	4	5	6	7	8	9	10
$E_i$ (°)	16.24160	16.08722	14.69055	15.27025	16.66231	16.98885	14.90985	17.11870	15.58698	14.85888
$N_i$ (°)	50.57513	50.27706	49.41570	49.95616	49.72949	49.95986	50.42491	49.46988	49.61639	49.99393
$q_i$	7	1	5	4	4	3	3	6	3	3
$p_i$	1	2	1	2	1	2	1	2	1	2
$i$	Customers									
	11	12	13	14	15	16	17	18	19	20
$E_i$ (°)	14.74052	15.63217	15.36320	15.598478	15.30779	15.04006	15.22476	16.65998	15.99651	15.90839
$N_i$ (°)	50.15575	50.35722	50.43471	49.42432	50.73916	50.76432	49.42156	49.49256	49.84431	50.02656
$q_i$	7	1	1	7	1	5	1	6	3	5
$p_i$	1	2	1	2	1	2	2	2	1	1
$i$	Customers							Stores		
	21	22	23	24	25	26	27	28	29	30
$E_i$ (°)	14.4257	16.5257	16.4257	16.1565	16.3101	17.3520	17.3535	14.6359	16.2264	16.7487
$N_i$ (°)	49.7105	50.0569	50.8749	50.7268	49.6891	50.8844	50.4806	51.0259	49.3061	50.6544
$q_i$	3	5	1	1	3	5	3	-	-	-
$p_i$	1	2	1	2	1	2	1	-	-	-

**Table 1** The customers and stores GPS coordinates  $E_i$ ,  $N_i$ , numbers of delivered containers  $q_i$ , priorities  $p_i$

By running the  $VRPMW\_SOLVER\_Z.m$  function with above-given input parameters, an optimal solution which consists of 4 cycles and 3 paths is found and the sum of travel distances of all vehicles is minimal. All cycles and paths are shown in Figure 1. This means that the delivery of 97 containers to customers can be ensured by seven trucks, and the total distance traveled by vehicles distance is 1 839,585 km the sum of driving times (hh:mm:ss) of all seven trucks, including the service times for unloading goods, is 41:32:44. The list of customers on the cycle/path, numbers of delivered containers, priorities, total lengths, and durations can be found in Table 2.

If the priority in the delivery of goods is canceled, it is possible to reach it by canceling condition 15, which is achieved in the m-function by canceling all command line No. 20. Then, the optimal solution consists of 3 cycles and 3 paths. The total distance traveled by vehicles is reduced to 1 689,264 km, and the sum of driving times of vehicles is reduced to 39:06:09.

If frozen and chilled goods will be delivered separately, in the command on line No. 18, the condition  $p(i) \sim p(j)$  will be replaced by the condition  $p(i) > p(j)$ . Then the optimal solution consists of 3 cycles and 3 paths such that the total distance traveled by vehicles is increased to 2 010,762 km, and the sum of driving times of vehicles is increased to 44:04:14.

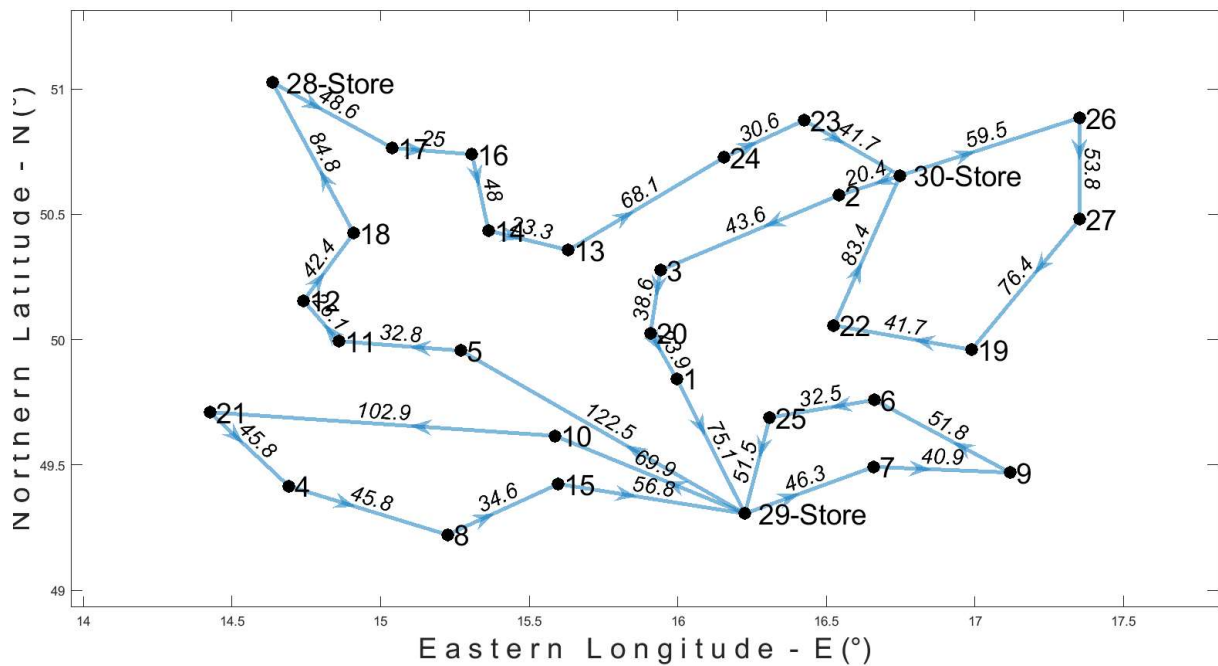


Figure 1 The optimal solution VRP for minimal traveling distance

No.		Start Store	Customers							Stop Store	Lengths (km)	Duration (hh:mm:ss)
1 Path	i	28	17	18	12	4	8	29	373.342	8:21:10		
	$q_i$	-	1	6	1	4	6	-				
	$p_i$	-	2	2	2	2	2	-				
2 Cycle	i	29	15	21	11	5	10	29	374.607	8:16:03		
	$q_i$	-	1	3	7	1	3	-				
	$p_i$	-	1	1	1	1	2	-				
3 Path	i	29	25	1	20	2	-	30	216.588	5:17:01		
	$q_i$	-	3	7	5	1	-	-				
	$p_i$	-	1	1	1	1	-	-				
4 Cycle	i	29	7	9	19	22	6	29	309.669	7:00:25		
	$q_i$	-	3	3	3	5	3	-				
	$p_i$	-	1	1	1	2	2	-				
5 Cycle	i	30	27	26	-	-	-	30	172.132	3:40:08		
	$q_i$	-	5	5	-	-	-	-				
	$p_i$	-	1	2	-	-	-	-				
6 Cycle	i	30	23	24	-	-	-	30	124.304	2:46:17		
	$q_i$	-	1	1	-	-	-	-				
	$p_i$	-	1	2	-	-	-	-				
7 Path	i	30	3	13	14	16	-	28	268.943	6:11:38		
	$q_i$	-	5	1	7	5	-	-				
	$p_i$	-	1	1	2	2	-	-				
								Total	1 839.585	41:32:44		

Table 2 The customers cycles and paths with respect priorities.

### 3 Conclusion

The set goal of developing an optimization model for solving the vehicle routing problem for multiple warehouses and prioritizing customers according to the temperature regime of the delivered goods was met. Optimization is realized by minimizing the sum of the distances traveled by all vehicles used to deliver goods. The completion of the created model is a software application, i.e. the creation of the M-function *VRPMW\_SOLVER\_Z.m*, which realizes the solution of the VRPMW problem for an arbitrary number of  $n$  customers and  $s$  warehouses. The created software was applied to an illustrative task with 3 warehouses and 27 customers. Its solution took about 15 minutes on a common laptop. The software is practically usable for up to 40 customers, but the processing time is increasing

rapidly. The application of the created software in logistics practice enables the saving of vehicle distances traveled and thus the fuel consumed.

## Acknowledgements

The paper was supported by the grant No. CZ.01.1.02/0.0/0.0/21\_374/0027244 "Technology development for intelligent traffic flow management - 2nd part - optimization and extension" of Czech Ministry of Industry and Trade.

## References

- [1] Eksioglu, B., Vural, A.V. & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57, 1472-1483.
- [2] Jonak, R., Smutný, Z., Simunek, M. & Dolezel, M. (2020). Rout and Travel Time Optimization for Delivery and Utility Services. *Acta Informatica Pragensia*, (2) 9, 200-209.
- [3] Math Works. Inc. (2024). *Optimization Toolbox™. User's Guide*. Natick.
- [4] Winston, W. L. (1994). *Operations Research. Applications and Algorithms*. Duxbury: Duxbury Press.
- [5] Toth, P. & Vigo, D. (1998). *Exact algorithms for vehicle routing*. Boston: Kluwer Academic Publisher
- [6] Zahrádka, J. (2023) The Exact Solution of Vehicle Routing Problem by Mixed Integer Linear Programming in Matlab. In J. Zahrádka, *Proceedings of 41st International Conference on Mathematical Methods in Economics* (pp, 433-438). Prague: Czech Society for Operations Research.

## Appendix

```

1: function [X, y, TotLgth, TotDur] = VRPMW_SOLVER_Z(n, s, D, C, Q, q, p, m)
2: options = optimoptions('intlinprog', 'MaxTime', 3600, 'MaxNodes', 3000000);
4: A=zeros(n*n-n, (n+s)*(n+s)+n); Aeq=zeros(2*n+s, (n+s)*(n+s)+n); TotLgth=0;
5: k=0; for i=1:n for j=1:n
6: if i~=j k=k+1;A(k, (n+s)*(n+s)+i)=1;A(k, (n+s)*(n+s)+j)=-1; end; end; end
7: k=0; for i=1:n for j=1:n if i~=j k=k+1; A(k, (n+s)*(i-1)+j)=Q; end; end; end
8: for i=1:n*n-n for j=1:n if A(i, (n+s)*(n+s)+j)==-1 b(i,1)=Q-q(j); end; end; end
9: k=n*(n-1); for i=1:s for j=1:n
10: k=k+1;A(k, n*(n+s)+(i-1)*(n+s)+j)=Q;A(k, (n+s)*(n+s)+j)=-1;b(k,1)=Q-q(j);end;end
11: for i=1:n for j=1:n+s Aeq(i, (i-1)*(n+s)+j)=1; end; beq(i,1)=1; end
12: for i=1:n for j=i:n+s:(n+s)*(n+s) Aeq(n+i,j)=1; end; beq(n+i,1)=1; end
13: for i=1:s for j=1:n
14: Aeq(2*n+i, (n+s)*n+(i-1)*(n+s)+j)=1; Aeq(2*n+i, (n+s)*(j-1)+n+i)=-1;
15: end; beq(2*n+i,1)=0; end
16: lb=zeros((n+s)*(n+s),1); ub=ones((n+s)*(n+s),1);
17: for i=1:n ub(i+(n+s)*(i-1), 1)=0; end
18: for i=1:s for j=1:s ub((n-1)*(n+s)+n+i*(n+s)+j, 1)=0; end; end
19: for i=1:n lb((n+s)*(n+s)+i, 1)=q(i); ub((n+s)*(n+s)+i,1)=Q; end
20: for i=1:n for j=1:n if p(i)>p(j); ub((n+s)*(i-1)+j)=0; end; end; end
21: ub=ub'; lb=lb'; DT=D';
22: f=DT(:);f((n+s)*(n+s)+1:(n+s)*(n+s)+n)=1; intcon=1:(n+s)^2+n;
23: intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, [], options);
24: V=round(V); X=V(1:(n+s)*(n+s));
25: X=reshape(X, [n+s, n+s]); X=X'; y=V((n+s)*(n+s)+1: end) ; TotDur=sum(m);
26: for i=1:(n+s) for j=1:(n+s) if X(i, j)==1
27: TotLgth=TotLgth+D(i, j); TotDur=TotDur+C(i, j); end; end; end
28: end

```