

Project Assignment 2 (v1.0) **Vehicle Routing Problem using Tabu Search**

Excerpt from the course syllabus in the study guide:

“The course aims to give the students an ability to model optimization problems, and an insight in how mathematical theory can be used to formulate and solve practical problems, with emphasis on applications in supply chain, distribution and transportation planning. The course also aims to give a deeper knowledge about combinatorial optimization, i.e. optimization problems with an underlying graph structure.”

This assignment tries to meet these aims by practicing your ability to :

- develop specialized methods for large-scale practical supply chain problems,
- implementation using a standard programming language.

In this assignment you should develop a meta-heuristic for an advanced Vehicle Routing Problem with several depots and hundreds of customers. In order to avoid local optimas, you need to develop a Tabu search heuristic.

Please hand in a printed copy of the report, as well as submitting it in Lisam.
Provide all MATLAB code in an appendix. (No need to print this part though.)

Deadline: December 2nd 2015

1 Background

Vehicle routing planning, based on optimization methods, can be used in order to significantly reduce transportation costs. Transportation is a necessary activity in the supply chain as it can ensure that goods are in the right place at the right time, where point of consumption not necessarily is the same as the point of production. In this sense transportation adds value in the supply chain. However, in which order, and by which vehicles a certain customer is served, does not necessarily change that added value; and thus it is reasonable to minimize costs; so as to minimize unnecessary parts of value adding activities, and make most efficient use of resources. For example, with a profit margin of 5%, reducing cost with 1%, equals the effect of increasing sales by 20%; which is why cost reduction may have a large effect on a company's net result.

In practice there may be many considerations where a daily or regular vehicle routing planning is not needed or even wanted; as there can be advantages, not necessarily measured in costs, of having e.g., fixed routes. However, solving a vehicle routing problem, that describes an alternative distribution scheme, can indicate at which cost non-monetary advantages are met.

1.1 Company description

Statoil (www.statoil.no, and www.statoil.se for the Swedish subsidiary) is a company active in the Energy industry. In this project, you will study a part of their Supply chain, looking at the distribution of gas to gas stations in Sweden. Statoil serves around 500 gas stations. Gas is transported from 11 depots.

2 Project

In this project you will be given a distribution problem, where distribution should be made to a subset of Statoil's customers, served by a subset of their depots.

2.1 Distribution problem

There is of course a number of simplifications made in this project, compared to Statoil's true distribution situation. The demand for each station is measured in integer cubic meters, the same unit as for the capacity of the vehicles used. The actual demands have been randomized. Further, only one vehicle type is considered, and a station must get its full demand fulfilled. (In practice, this is not necessarily true. The dispatcher can negotiate to reduce delivered volume, in order to fit a particular customer in a specific route. Or the truck driver may deliver more than the demand, if there is some extra gas in the truck, not booked by any other customer.)

In reality, there are different qualities of fuel (gas/diesel/etc.). A vehicle normally contains several separated compartments; and what limits the total capacity of the truck, is often the weight rather than the volume. Thus a truck can usually take any combination of demands of different qualities/commodities; and it will not be a severe simplification to consider only total demand, total capacity (which we measure in volume), and a single commodity.

There is only one type of vehicle available in the scenario, i.e., the fleet is homogeneous. It is reasonable to assume that there is a sufficient (or large) number of vehicles of this type available at each depot. In practice, if a solution requires more vehicles than physically available, some vehicles can be scheduled to make several routes on the same day. We can also disregard the maximum length on the routes, as in practice routes will normally be covered by one (possibly long) shift.

There are multiple depots, and a customer may be served from any depot. Split delivery is not permitted, thus each customer can only be visited by one vehicle (per day). Only one day is considered. Thus, we have a multi-depot, homogeneous vehicle routing problem to solve. The problem instance is illustrated in Figure 1, with a total of 7 depots and 237 customers all over the south of Sweden.

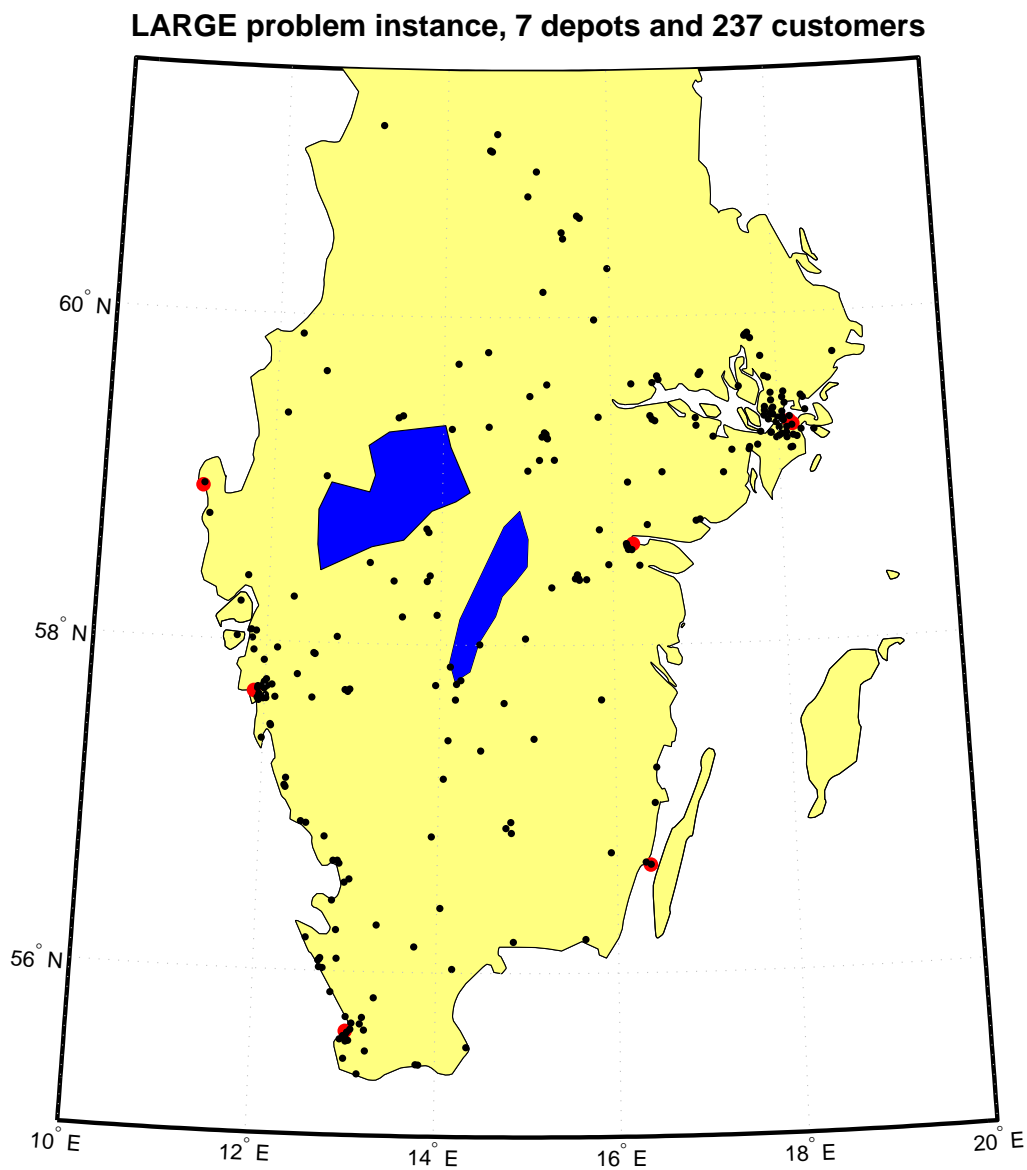


Figure 1: Illustration of the big problem instance. The 7 depots are shown as red circles and the smaller black dots represent each of the 237 customers.

2.2 Problem parameters

There is a number of parameters to be used. One set of parameters relates to the available vehicle type, and one set relates to the Depots and Customers.

2.2.1 Vehicle parameters

We assume a homogeneous fleet of vehicles with a given capacity. There is both a fixed cost that must be paid each time the vehicle is used, as well as a mileage cost proportional to the traveled distance. There is also a stop cost for each customer visited. The parameter values are given in Table 1.

Table 1: Specification of the vehicle parameters

Parameter	Capacity (m3)	Fixed Cost (SEK/route)	Mileage Cost (SEK/km)	Stop Cost (SEK/customer)
Value	48	300	12	40

Some remarks can be made regarding the costs. In a mathematical formulation of the problem, both the fixed cost for a route and the mileage cost could be transformed into a cost matrix, based on the distance matrix. Although, when designing a heuristic, it might be just as easy to compute the total costs using the parameters in Table 1.

2.2.2 Depot and Customer parameters

The location of the depots and the customers are defined by coordinates, given in the separate MATLAB files (see Lisam), along with a distance matrix. Distances are measured in kilometers, based on actual road distances, and the distance matrix is symmetric and fulfills the triangle inequality. Each customer has a demand, and each depot has a limited supply. The problem instance is illustrated in Figure 1.

2.3 Problem data

All data is specified in the provided MATLAB files. There are two problem instances available, `SMALL.mat` and `LARGE.mat`, and it is recommended to use the former one while developing and testing your heuristic. You are also encouraged to generate your own data, as it might be helpful to test your code on specific problem instances where you can see if a particular solution is reasonable.

2.4 Solution method

The method to use in order to solve the problem is a Tabu search. There are of course a number of choices to make. One important part of the project is to reflect over alternative implementations, and to motivate the choices you make, and possibly to try several alternatives and choose the most promising one. This includes the choice of representation of the problem, and which information that needs to be stored in

the solution process. It also includes how to create an initial (probably feasible) solution from which the Tabu search procedure can make improvements.

Further, one also needs to define and motivate the choice of Neighborhood, i.e., which moves/changes/switches that can be made in one iteration. Note that this does not necessarily have to be limited to only one particular move; it can include different types of moves, either in each major iteration, or in a sequence. It includes how, based on the Neighborhood definition, moves are considered Tabu. In the development phase, this probably means that it is wise to try to identify when cycling occurs (which may indicate that there are too few solutions that are Tabu).

Furthermore, it includes the choice of how long the Tabu list should be, and of course also which other refinements of Tabu search that should be implemented. Also one (or several) stopping criteria must be chosen. Possibly the project include a number of other choices as well.

See also the specific assignments in Section 4 of this document.

2.5 Programming environment

The MATLAB file contains all data needed to define the problem. It also contains a function that can be used to graphically illustrate the problem (locations of all depots and customers) and also to draw solutions.

The intention is that the implementation should be done in MATLAB, but you may choose another programming language if you like. Tutoring on the programming level can not be expected, so you need to clearly define your Tabu procedure.

A Tabu search heuristic can be constructed without too much advanced coding. The basic ingredients in such a code will be arithmetic operations on matrices, vectors and scalars, For- statements and If-statements. This means that any basic programming skills should be adequate.

MATLAB is a so called interpreted language, and hence no compilation of the code is needed. In general, if you know any programming language, MATLAB is straightforward to use.

3 Project report

The target group for the report is the Distribution Manager of Statoil A/S, i.e., the Norwegian mother company. She took an Operations Research course some 20 years ago, so although well aware of the concepts of their distribution, you need to briefly explain the particular distribution problem you've been solving. Further, you need to briefly explain the general principles behind a Tabu search, since that was a very new concept at the time of her studies.

She may distribute the report to the Board as well, so it needs to be well written. However, the focus of the report should of course be to clearly describe and motivate the design and details of your Tabu search heuristic, and comment on the behavior of the solution process, as well as the results. This should include answers or comments or descriptions relating to the assignments of Section 4, but preferably woven into the report, and not necessarily as separate answers to questions (as the Board, in particular, neither will have access to nor be interested in the particular questions; but rather on the related discussions).

4 Assignments

An overview of the assignments is given in Table 2. The points for each assignment is supposed to correspond to the level of difficulty and required time.

Table 2: Overview of assignments

	1	2	3	4	5	6
Topic	Assign Customers	Constr. Heuristic	Basic Tabu	Compute a LBD	Implement Divers/Intens.	Allow Infeas.
Points	2	5	10	3	3 + 4	3

4.1 Assign customers to the depots (2p)

Based on the customer demands and limited depot supply, as well as the distances between the customers and the depots, assign each customer to one depot. Make sure that the solution is feasible with respect to the depot supply. What kind of optimization problem is this? Can you give a mathematical model?

This can be seen as a strategical planning step, and the assignment of customers to depots found here should be used throughout all the following project assignments. Hence our distribution problem is decomposed into one VRP-problem for each depot.

4.2 Constructive heuristic (5p)

Implement a constructive heuristic code in order to generate a feasible solution to the distribution problem. This is normally the first step in a Tabu search heuristic. Note the objective function value of the starting solution found by your constructive heuristic.

4.3 Basic Tabu search (10p)

Implement a basic Tabu search heuristic. As this is the key ingredient of the project, you should of course be able to clearly describe the components of the heuristic, and motivate all choices that are made. Comment on the computational effort of each solution in the neighborhood, related to the size of the neighborhood. Comment also on the scale of reduction of possible moves in the neighborhood, relating to the choice of the definition of what is Tabu.

It is critical that your Tabu search heuristic allows both for increasing as well as decreasing the number of routes used. A heuristic that keeps the total number of routes fixed at all times will not be considered acceptable.

Use your basic Tabu search heuristic to solve the provided instance **LARGE**. The best solution is of course of interest, but the iteration data is also of interest. Include a graph on the solution history of the heuristic, similar to the one in Figure 2. For each iteration, keep record of the “Best found solution” and the “Current objective function value”. If an evaluation function different from the objective function is used, record also the “Current evaluation function value”.

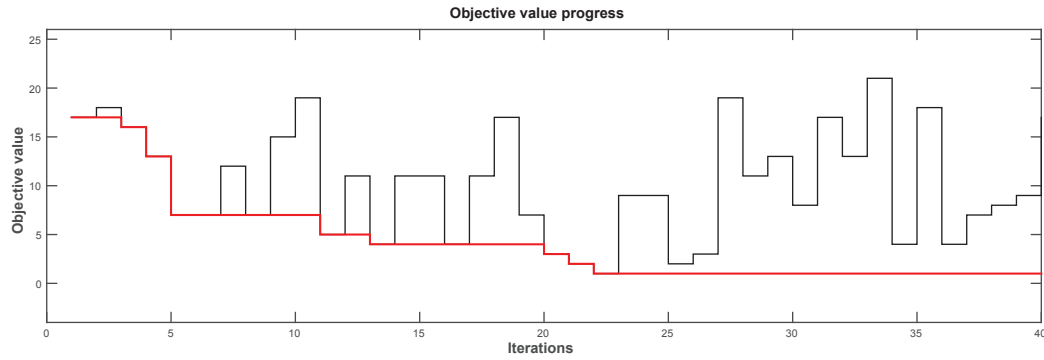


Figure 2: Illustration of the progress of the objective value. The black line illustrates how the current objective value fluctuates, and the red line is the best objective found so far.

Make sure to look out for a so called *cycling* behaviour in the progress, as illustrated in Figure 3, because this clearly indicates that the Tabu search is not functional. It can happen for example if the Tabu list is too short, or if the neighborhood used is too small.

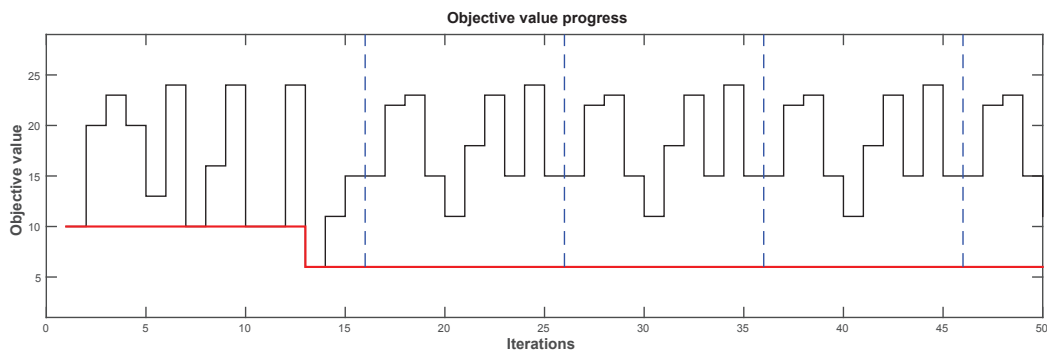


Figure 3: Illustration of a Tabu search progress stuck in a cycle. Here the vertical blue dashed lines indicate the repetitive cyclic pattern.

4.4 Compute a Lower bound (3p)

Implement some kind of lower bound. Of course it is better to implement a “good” lower bound, than a worse one, but the important part of this assignment is to discuss the principles of bounding. You should not base your lower bound implementation on relaxations of mathematical models, since that probably would require too much effort. The suggestion is that a lower bound is computed using observation of how many arcs, and at which costs, that at least must be used in an optimal solution.

It is OK to have a relative gap as large as 90% (or even more) between the solution found in the Tabu search heuristic and the lower bound, as long as the discussion around the bounding technique is relevant. For given upper and lower bounds, denoted UBD and LBD, the relative gap is defined as $(UBD - LBD)/LBD$.

4.5 Implement Diversification & Intensification

A simple diversification (3p)

A naïve way of diversifying is to restart from a/some new initial solution(s). Add this feature to your heuristic and evaluate if the diversification strategy improves the result. This implies that some kind of progress measure is needed in order to know when to perform the diversification.

Intensification and/or advanced diversification (4p)

Implement an intensification strategy and/or an diversification strategy which makes use of long term memory (e.g., frequency and/or recency, in some sense). Evaluate if the implemented strategy improves the best solution. If it doesn't, be sure to give a motivation why you think it failed.

4.6 Allow moves to infeasible solutions (3p)

Implement some way of allowing the solution process to (temporarily) visit infeasible solutions (e.g., where a route has a total demand larger than the capacity of the largest vehicle, or some customer not being served). This would mean that another function is used to evaluate moves. Be sure to tune that procedure, so that it is likely that the solution procedure will return to feasible solutions (which usually means that solutions are punished more, the “more infeasible” they are and/or for how long they have been infeasible).

5 Analysis of the results

Make sure to analyze the best routings found by your Tabu search. This analysis should be done for each individual depot, as well as for the full distribution problem. Besides the objective value, also look at interesting measures and features such as:

- How many routes and trucks are used?
- How many customers are served on average on each route?
- What is the minimum/maximum number of customers on a single route?
- How long is the shortest/longest route?
- How long are routes on average?

You are also encouraged to investigate how much the best found solution can be improved upon if the truck capacities are increased to (for example) 60 units.

Further, reflect upon the initial assignment of customers to depots, is it possible to introduce moves that allow for customers to be assigned to a new depot?

6 Matlab

The provided data files, `SMALL.mat` and `LARGE.mat`, contain all data needed to define the two problem instances. The data is stored in a so called *struct* object, and to get access to the struct one uses the command `load`, see an example below.

The small problem instance contains data for 3 depots and 25 customers, and its data struct is illustrated below. Besides a number of scalar values, there is a vector for the customer demands and matrices that contain distances between all pairwise locations (both depots and customers).

```
>> load('SMALL');

>> SMALL

SMALL =
    nrDepo: 3
    nrCust: 25
    Capacity: 48
    FixedCost: 300
    StopCost: 40
    MileageCost: 12
    Coord: [1x1 struct]
    Demand: [25x1 double]
    Supply: [120 60 120]
    Dist: [1x1 struct]

>> SMALL.Demand'

ans =
Columns 1 through 13

    10    20    30    20    28    22     3     6     7    10     6    10    10

Columns 14 through 25

     5     2     4    10     7     8    14     6    11     8     3    12

>> SMALL.Dist

ans =
    d2d: [3x3 double]
    d2c: [3x25 double]
    c2c: [25x25 double]

>> SMALL.Dist.d2c

ans =
Columns 1 through 13

    173    165    108    151    160    139    186    177    175    122    150    119    127
    202    226    163    201    166    166     88     92     87    136     85    123    108
     34     40    102     50     64     78    142    136    141    112    149    127    138

Columns 14 through 25

    168    186    149    106    127     65     57     61     64    106    133    107
     54     36     71    123     90    180    190    161    151    106     76    103
    177    195    169    139    166    140    144    157    163    188    202    197
```

6.1 Plots

In order to visualize the problem instances, as well as illustrate solutions, a plot function is available.

```
>> plotProblem(SMALL);
```

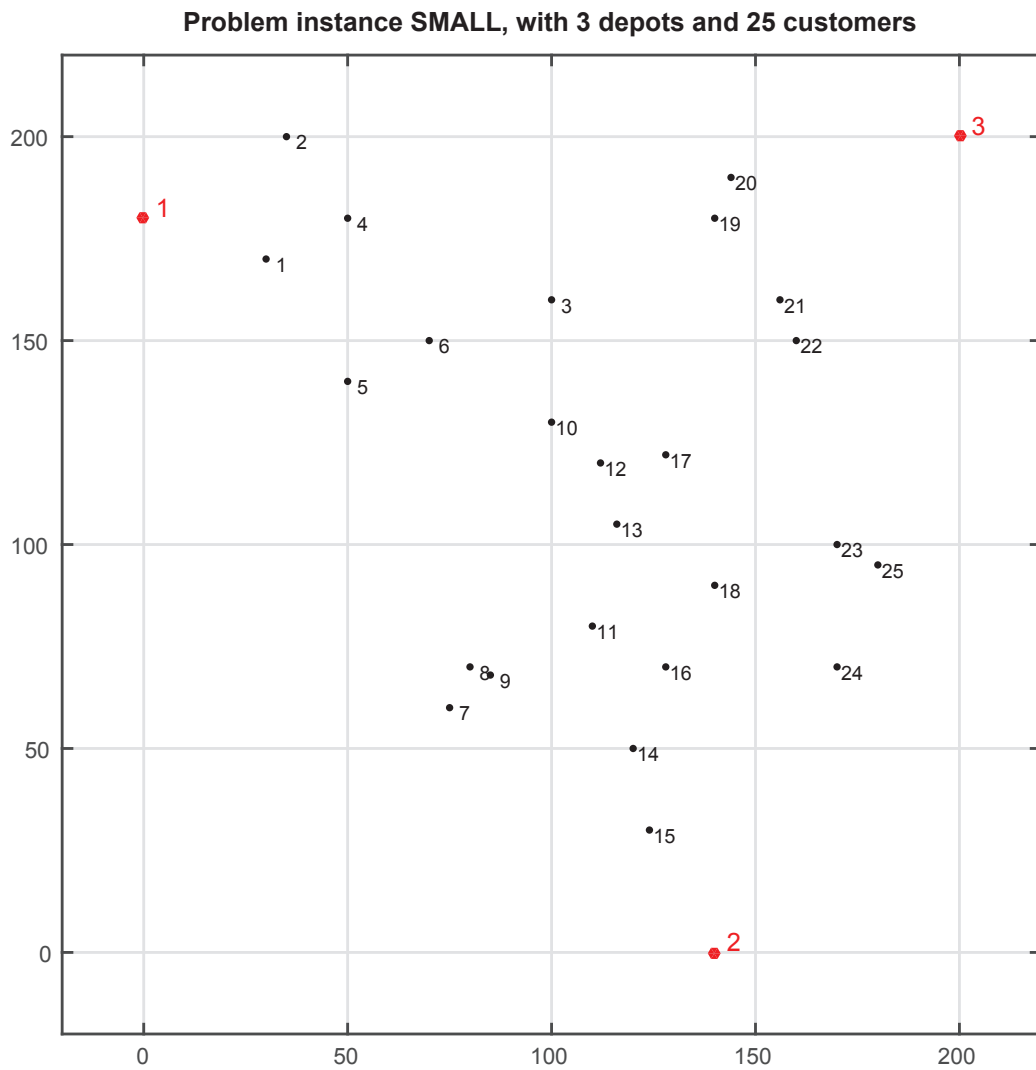


Figure 4: The problem instance SMALL contains 3 depots and 25 customers.

One way of representing a route is to use a so called *cell* object, a generalization of a vector. The following lines of MATLAB code creates a cell parameter and assigns four routes. One could for example use negative numbers to indicate depots.

```
>> ROUTE = cell(1,4);
>> ROUTE{1} = [-1 2 4 6 5 1];
>> ROUTE{2} = [-2 15 7 8 9 11 16 14];
>> ROUTE{3} = [-2 24 25 23 18];
>> ROUTE{4} = [-3 20 19 21 22];
>> plotSolution(SMALL,ROUTE);
```

The first route starts from depot 1 and visits customers 2–4–6–5–1, and then goes back to the depot. Second route starts from depot 2 and visits customers 15–7–8–9–11–16–14, and then goes back to the depot. Second route also starts from depot 2 and visits customers 24–25–23–18, and then goes back to the depot. The fourth route starts from depot 3 and visits customers 20–19–21–22, and then goes back to the depot.

This partial solution (all customers are not covered) is illustrated in Figure 5.

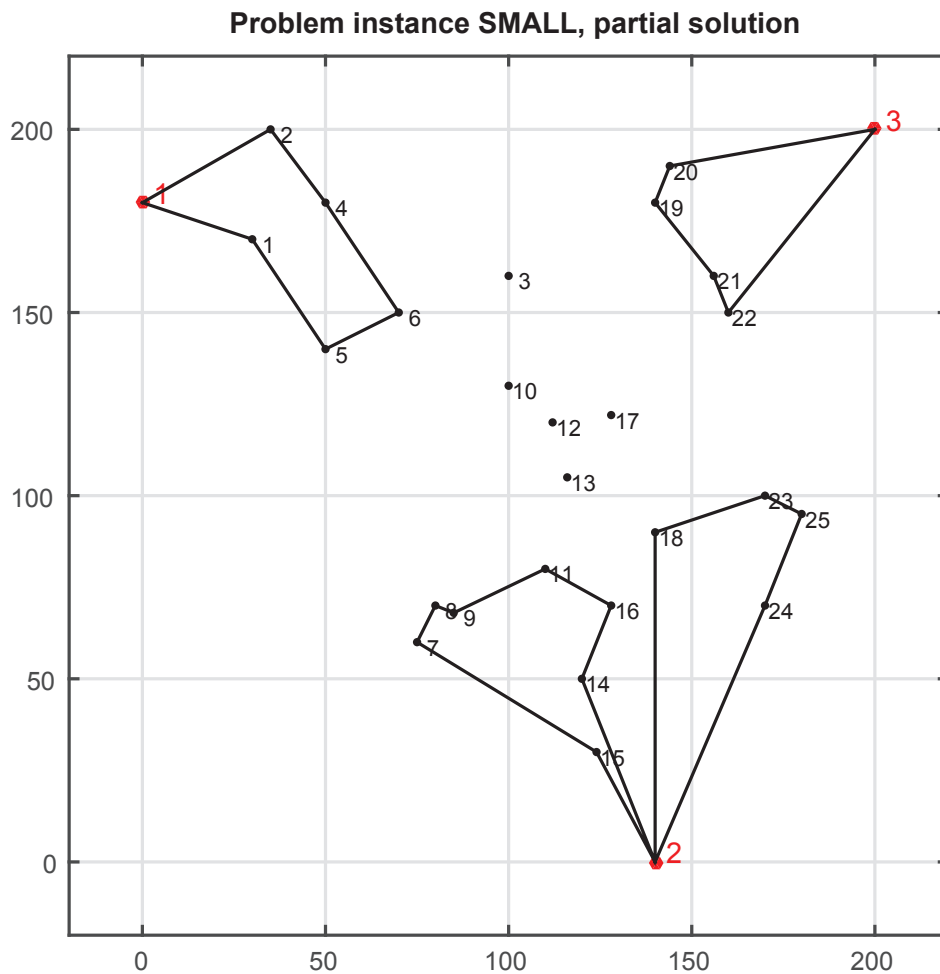


Figure 5: A partial solution to the small problem instance.

6.2 Customer distances

```
>> SMALL.Dist.c2c
```

```
ans =
```

```
Columns 1 through 13
```

0	30	71	22	36	47	119	112	116	81	120	96	108
30	0	76	25	62	61	146	138	141	96	142	111	125
71	76	0	54	54	34	103	92	93	32	81	42	57
22	25	54	0	40	36	123	114	117	71	117	86	100
36	62	54	40	0	24	84	76	80	51	85	67	75
47	61	34	36	24	0	90	81	83	38	81	52	64
119	146	103	123	84	90	0	13	13	74	40	70	61
112	138	92	114	76	81	13	0	7	63	32	61	50
116	141	93	117	80	83	13	7	0	64	28	59	48
81	96	32	71	51	38	74	63	64	0	53	16	30
120	142	81	117	85	81	40	32	28	53	0	40	26
96	111	42	86	67	52	70	61	59	16	40	0	16
108	125	57	100	75	64	61	50	48	30	26	16	0
150	172	112	148	114	112	46	43	41	82	34	70	55
169	192	132	167	133	132	57	59	54	103	52	91	75
140	160	94	135	105	99	54	48	43	66	21	52	37
109	121	47	97	80	66	82	71	69	31	46	16	21
136	152	81	127	103	92	72	63	59	57	32	41	28
110	107	47	90	98	76	136	125	125	64	104	66	79
116	109	53	95	106	84	147	136	136	74	115	77	89
126	127	56	108	108	87	129	118	116	64	92	61	68
132	135	61	114	110	90	124	113	111	63	86	57	63
157	168	92	144	126	112	103	95	91	76	63	61	54
172	187	114	163	139	128	96	90	85	92	61	77	64
168	179	103	155	136	123	111	103	99	87	72	74	65

```
Columns 14 through 25
```

150	169	140	109	136	110	116	126	132	157	172	168
172	192	160	121	152	107	109	127	135	168	187	179
112	132	94	47	81	47	53	56	61	92	114	103
148	167	135	97	127	90	95	108	114	144	163	155
114	133	105	80	103	98	106	108	110	126	139	136
112	132	99	66	92	76	84	87	90	112	128	123
46	57	54	82	72	136	147	129	124	103	96	111
43	59	48	71	63	125	136	118	113	95	90	103
41	54	43	69	59	125	136	116	111	91	85	99
82	103	66	31	57	64	74	64	63	76	92	87
34	52	21	46	32	104	115	92	86	63	61	72
70	91	52	16	41	66	77	61	57	61	77	74
55	75	37	21	28	79	89	68	63	54	64	65
0	22	24	74	45	132	140	116	108	71	54	75
22	0	40	92	62	151	161	134	125	84	61	86
24	40	0	52	25	111	121	94	86	52	42	58
74	92	52	0	34	61	70	47	43	49	67	59
45	62	25	34	0	90	100	72	63	32	36	40
132	151	111	61	90	0	13	26	36	85	114	94
140	161	121	70	100	13	0	34	43	94	123	100
116	134	94	47	72	26	34	0	13	62	91	69
108	125	86	43	63	36	43	13	0	51	81	61
71	84	52	49	32	85	94	62	51	0	30	13
54	61	42	67	36	114	123	91	81	30	0	27
75	86	58	59	40	94	100	69	61	13	27	0