Aalto University
School of Science

The Distance-Constrained Generalized Vehicle Routing Problem

Markus Mattila

7.6.2016

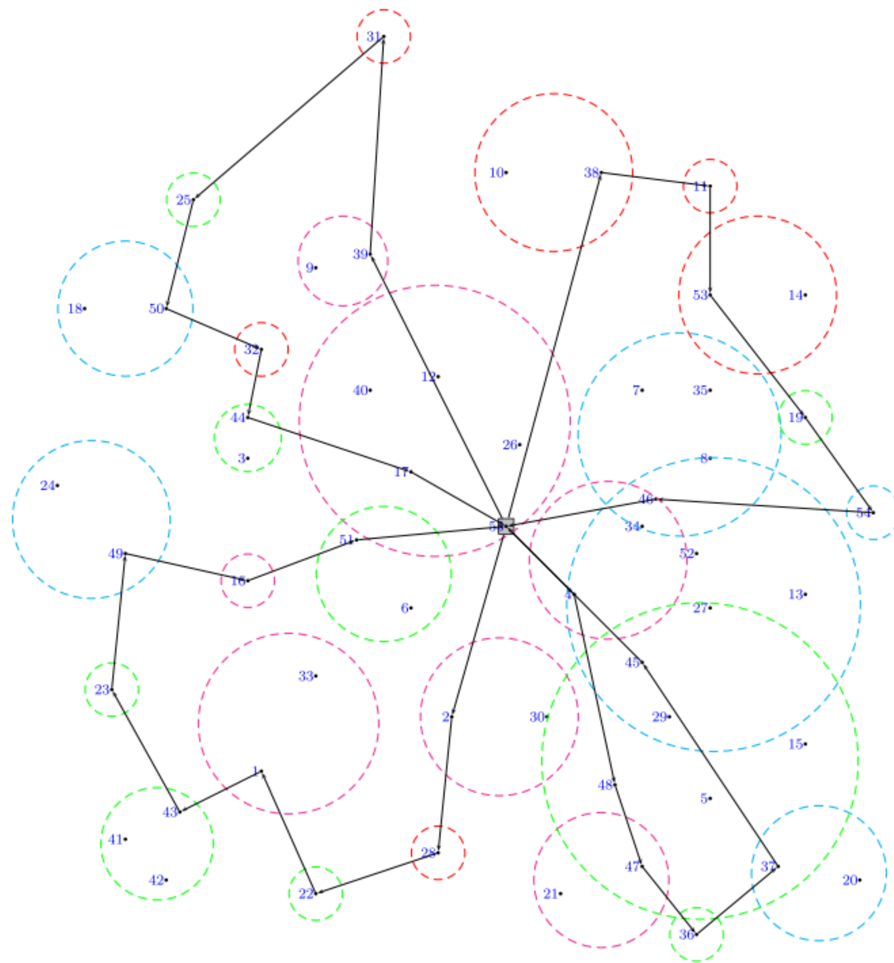
Presentation Structure

Problem description and formulations

Motivation

Heuristic algorithm

Results



Problem description

Generalization of the Travelling Salesman Problem (TSP)

We have a set of *n customers* and connecting *arcs*

Total route length must be minimized

$$\min \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} d_{ij} x_{ij}$$

Problem description

Further generalization: Capacitated Vehicle Routing Problem (CVRP)

Vehicles have a capacity *Q* and customers a demand q each

Fleet consists of *m* vehicles

$$\sum_{j=1}^{n} x_{0j} = m$$

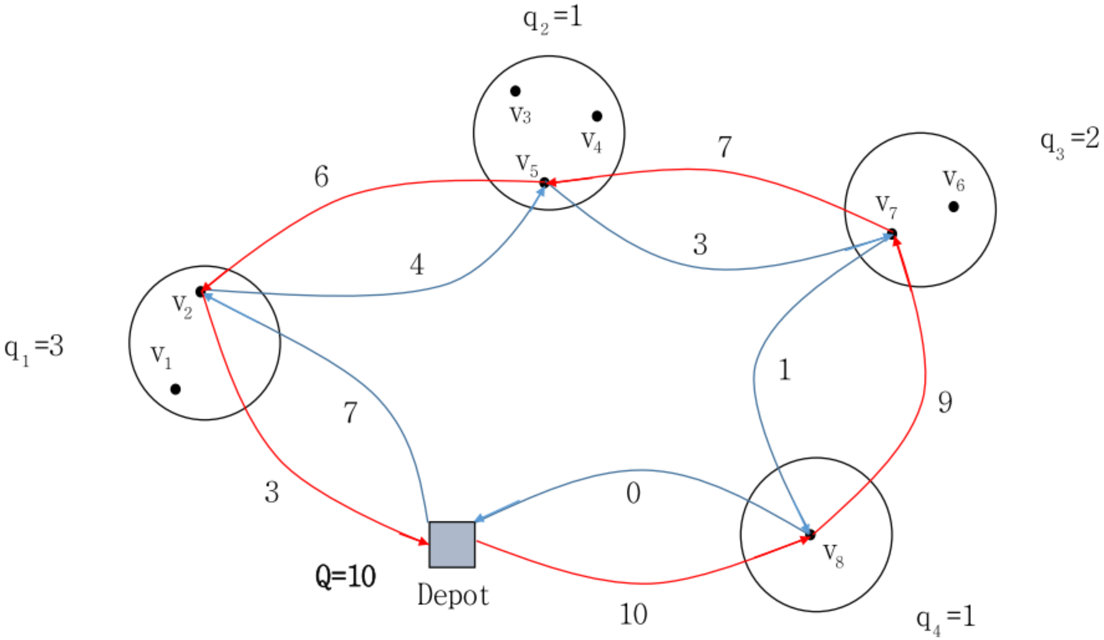
***m* can be fixed or variable**

$$\sum_{j=1}^{n} x_{jn+1} = m$$

Problem description

We use a two commodity flow model for capacity constraints

Forward flow describes the amount of goods in a vehicle and backward flow the amount of free space



$$f_{ij} + f_{ji} = Q(x_{ij} + x_{ji})$$

$$\sum_{j=1}^{n} f_{n+1j} = Qm$$

$$\sum_{j=1}^{n} f_{0j} = \sum_{k=1}^{K} q_k$$

$$\sum_{j=0}^{n+1} f_{ji} - \sum_{j=0}^{n+1} f_{ij} = 2\tilde{q}_i y_i$$

Problem description

Further generalization: Generalized Vehicle Routing Problem (GVRP)

Customers are divided into *K* clusters

$$\sum_{i \in C_l} y_i = 1$$

Exactly one customer per cluster must be visited

$$\sum_{i=0}^{n} x_{ij} = y_j$$

$$\sum_{k=1}^{n+1} x_{jk} = y_j$$

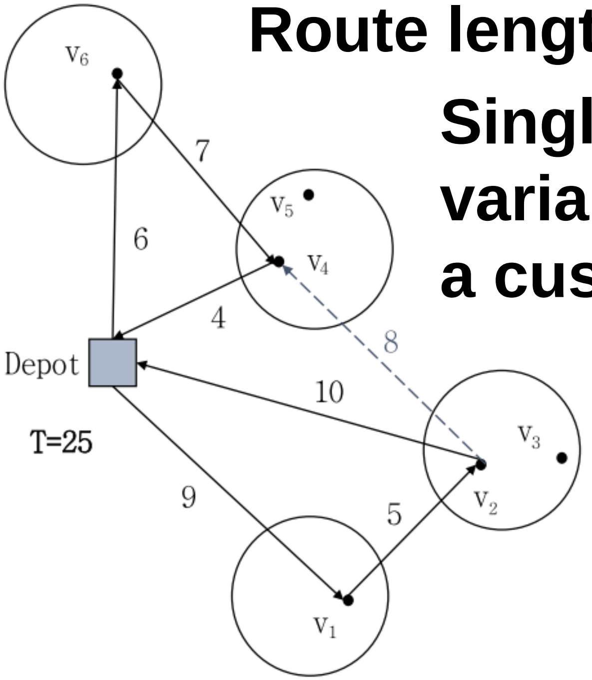
Problem description

Further (and final) generalization:

Distance-Constrained Generalized Vehicle Routing Problem (DGVRP)

Route length is constrained to an upper limit *T*

Single commodity flow formulation: variable values describe the arrival time at a customer after traversing an arc



$$\sum_{j=0}^{n+1} h_{ij} - \sum_{k=0}^{n+1} h_{ki} = \sum_{j=0}^{n+1} t_{ij} x_{ij}$$

$$h_{ij} \leq T x_{ij}$$

$$h_{0i} = t_{0i} x_{0i}$$

Motivation

Adding distance constraints allows creating more realistic models for real-life applications

Disaster aid – delivering supplies airborne to interconnected villages

- **Distance constraints modelling fuel capacity**

Designing routes for vehicles delivering industrial products

- **Constraints modelling the length of a work shift**

Motivation

Many problems, such as the School Bus Routing Problem and the Capacitated m-Ring-Star Problem can be transformed to a DGVRP

The distance constraints in the transformed DGVRP can model a variety of things that may be difficult to implement to the original problem

Many possible applications have not been considered

Heuristic algorithm

The DGVRP problem size grows exponentially as more customers are added

Real-life scenarios tend to be huge

➡ In stead of solving the problem to optimality, we search for good solutions heuristically

Heuristic algorithm

```
for i ← 1 to n_i do
    Routes ← InitialRoutes(i)
    Routes ← LocalSearch(Routes)
    GiantTour ← Concat(Routes)
    for j ← 1 to n_m do
        MutatedTour ← Mutate(GiantTour)
        Routes ← Split(MutatedTour)
        Routes ← LocalSearch(Routes)
    end
end
Solution ← SetPartitioning(StoredRoutes)
```

Heuristic algorithm: local search

Our local search consists of three ’classic’ moves:

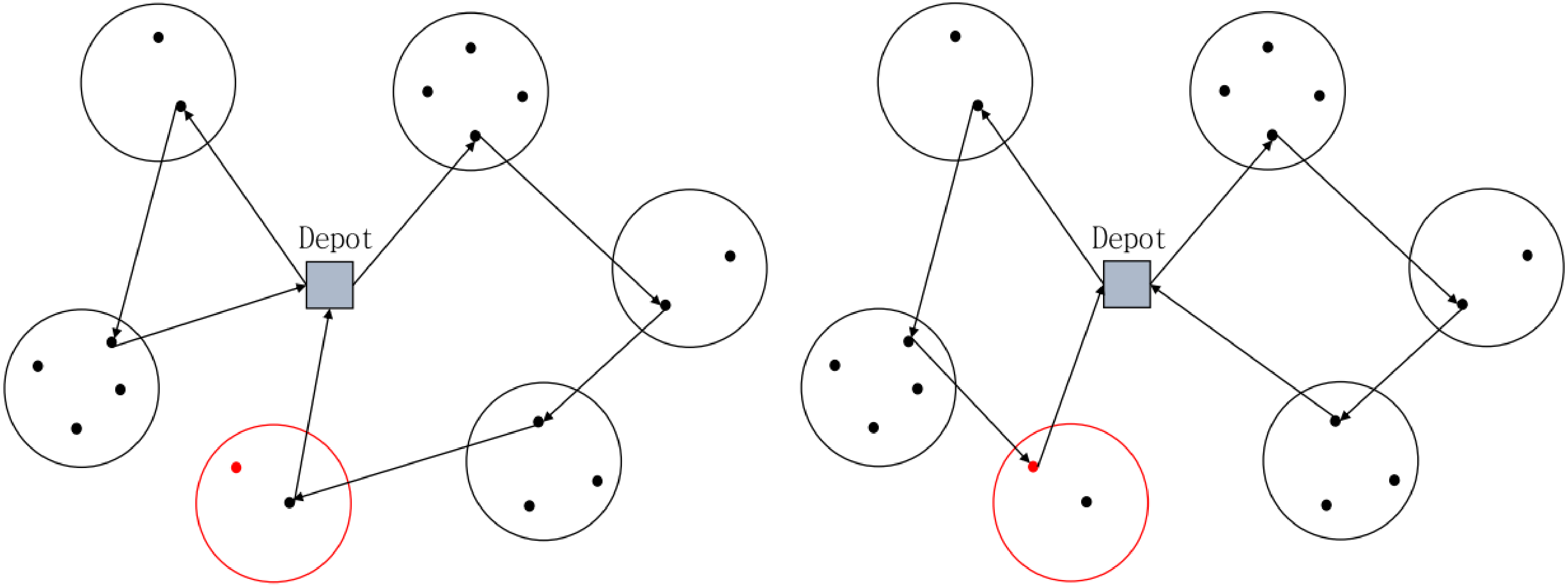
One-point (relocation)

Two-point (swap)

2-opt

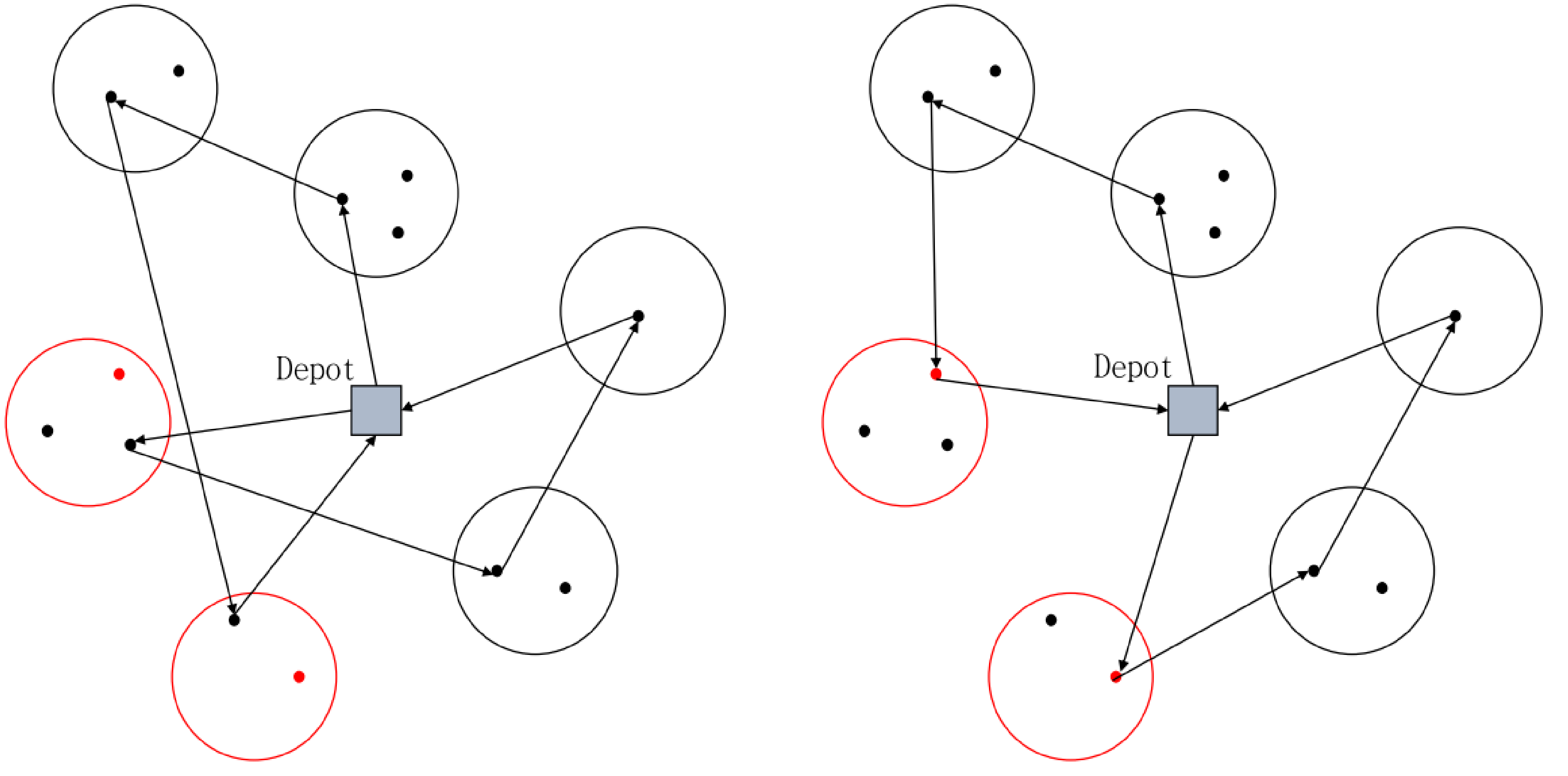
Heuristic algorithm: local search

One-point



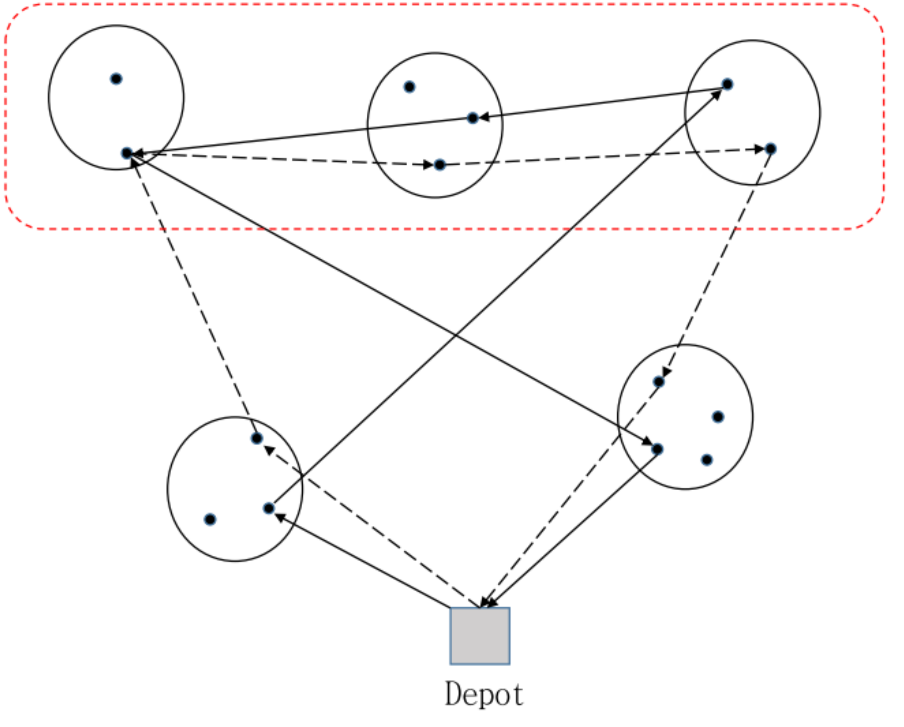
Heuristic algorithm: local search

Two-point



Heuristic algorithm: local search

2-opt



Heuristic algorithm: concat & split

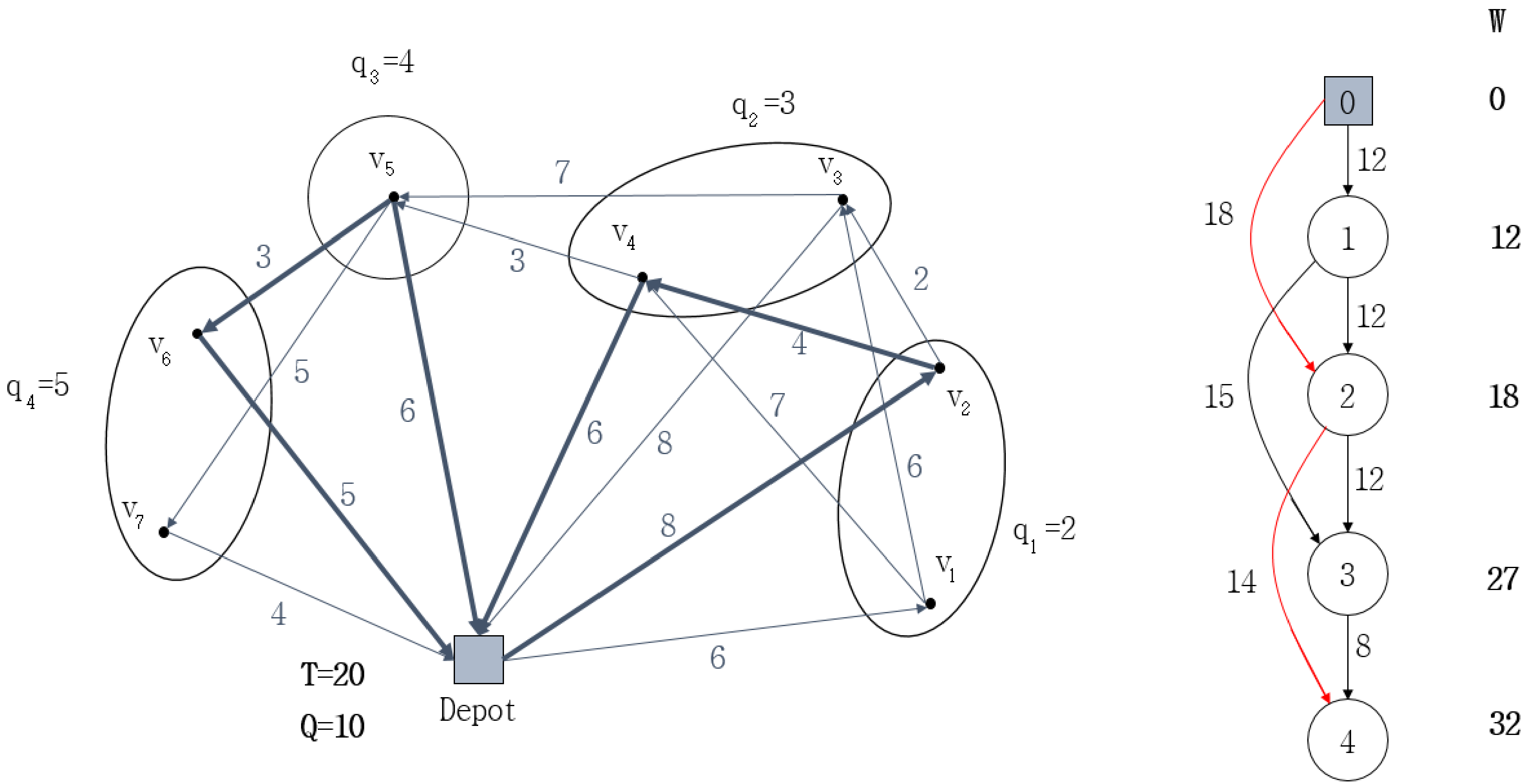
After the local search, we have a locally optimal solution

To move to a new neighbourhood, we first merge the routes into a single tour covering all clusters

The giant tour is mutated and split into feasible routes

Having the order of the clusters fixed, the split algorithm yields an optimal set of routes in reasonable computing time

Heuristic algorithm: concat & split



Heuristic algorithm: set partitioning

Each time after the local search or split procedure, we store the locally optimal routes.

After the desired number of iterations, we have a pool of ’good’ routes

We solve a set partitioning problem, combining routes so that each cluster is visited exactly once in the final solution

The procedure is quick and improves solution quality considerably.

Results

The heuristic algorithm was tested on available GVRP instances (without distance constraints)

The algorithm was executed five times

Table 1: Summary and comparison of the algorithm performance on small and medium instances

Instance set	θ	Mattila	Bektaş et al.	Moccia et al.	Afsar et al.	Hà et al
A	2	**27/27**	**25/25**	**25/25**	**27/27**	**27/27**
B	2	**23/23**	**23/23**	21/23	20/23	**23/23**
P	2	23/24	22/24	20/24	22/24	**24/24**
A	3	**27/27**	25/26	25/26	**27/27**	**27/27**
B	3	**23/23**	22/23	**23/23**	**23/23**	**23/23**
P	3	**24/24**	21/24	23/24	23/24	**24/24**

Results

For big instances (*n > 100),* the algorithm performance is relatively weaker

Table 2: Heuristic algorithm result comparison for big instances

Instance	θ	LB	Mattila	Bektaş et al.	Moccia et al.	Hà et al
M-n101-k10-C51	2	542.00	**542**	**542**	**542**	**542**
M-n121-k7-C61	2	705.84	721	**719**	720	**719**
M-n151-k12-C76	2	634.65	**659**	**659**	**659**	**659**
M-n200-k16-C100	2	752.14	791	791	805	**789**
G-n262-k25-C131	2	2945.02	3285	**3249**	3319	3303
M-n101-k10-C34	3	458.00	**458**	**458**	**458**	**458**
M-n121-k7-C41	3	527.00	531	**527**	**527**	**527**
M-n151-k12-C51	3	474.32	**483**	**483**	**483**	**483**
M-n200-k16-C67	3	572.81	**605**	**605**	**605**	**605**
G-n262-k25-C88	3	2239.50	2486	2476	**2463**	2477

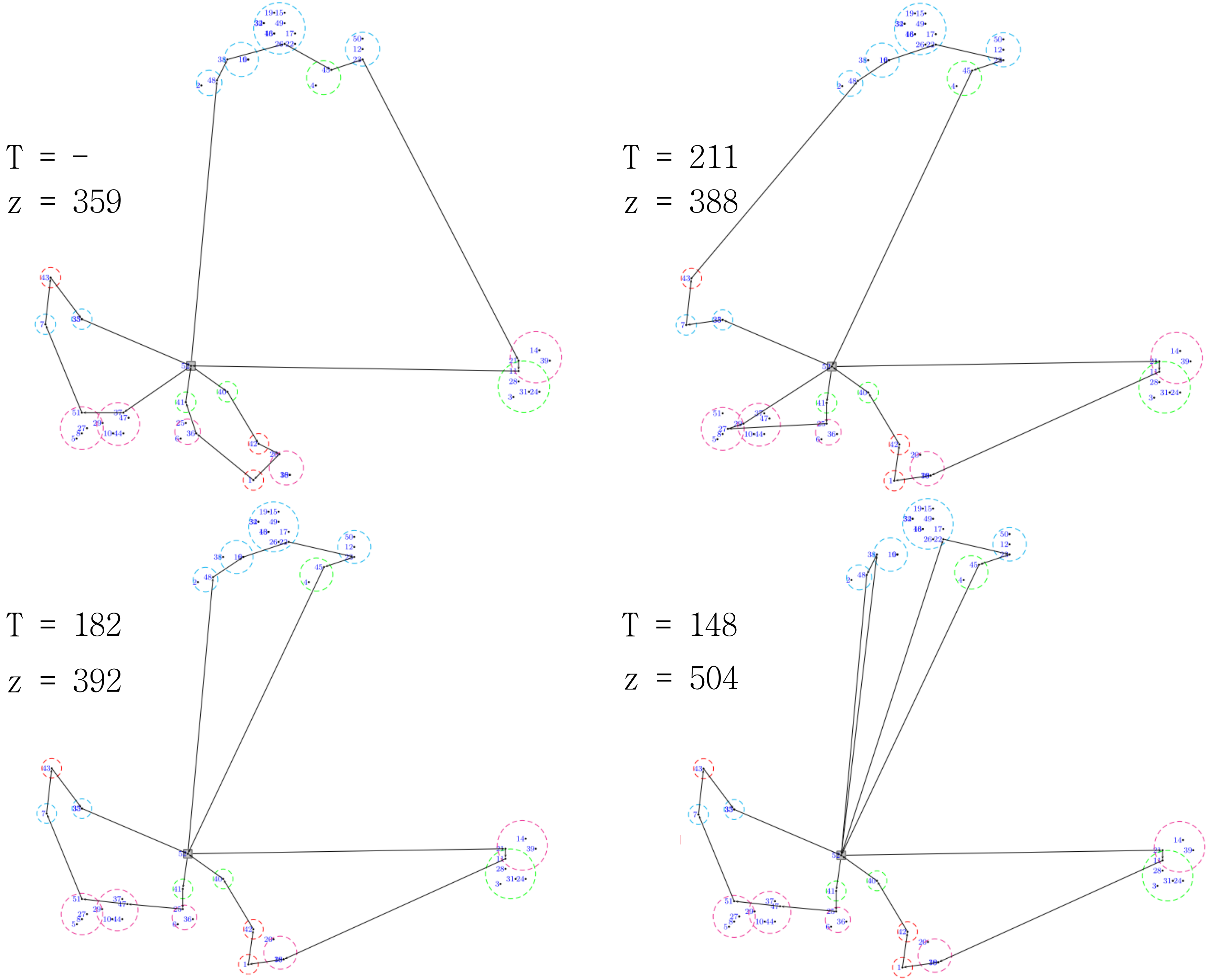
Results

To test the DGVRP, we derived our own instances from the GVRP instances

The limit was iteratively set lower to the length of the longest route in the previous solution

Lower bounds were calculated by a commercial solver CPLEX with a 2 hour time limit

The heuristic found all optimal solutions, and found at least as good results as CPLEX in all cases when the time limit was exceeded


T = -
z = 359
T = 211
z = 388
T = 182
z = 392
T = 148
z = 504

Aalto University
School of Science

Questions