UNIVERSITÄT PADERBORN

Metaheuristics for the vehicle routing problem

Osman‘s Simulated Annealing Algorithm‘s

Agenda

UNIVERSITÄT PADERBORN

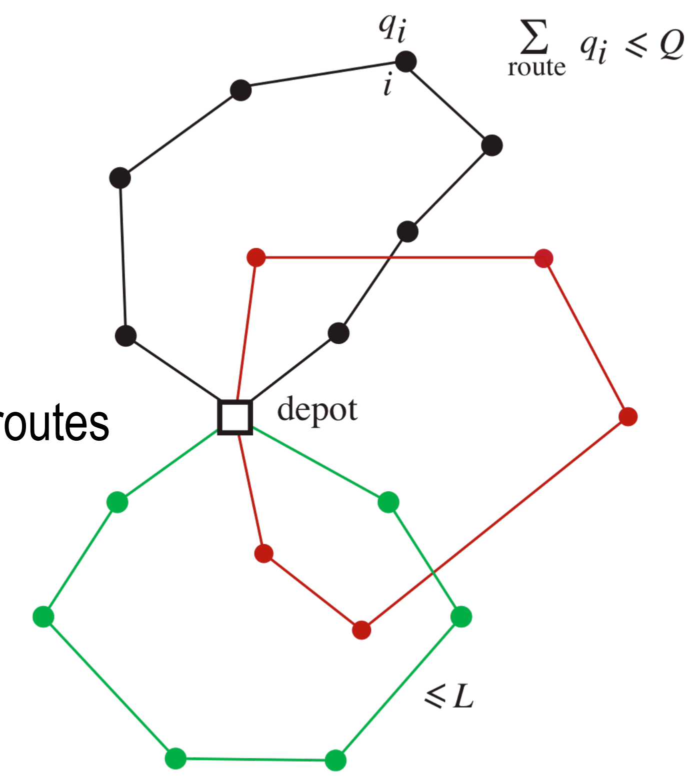
- VRP – The Vehicle Routing Problem
- Heuristics for routing problems
- Simulated Annealing
- Osman‘s SA
- Computational Experiments
- Questions

VRP – The Vehicle Routing Problem

- Problem introduced by Dantzig and Ramser (Management Science, 1959)
- NP-hard
- Generalization of the traveling salesman problem
- Has multiple applications
- Exact algorithms: relatively small instances
- In practice heuristics are used
- Variants:
  - heterogeneous vehicle fleet (Gendreau et al., 1999)
  - time windows (Cordeau et al., VRP book, 2002)
  - pickup and deliveries (Desaulniers et al., VRP book, 2002)
  - periodic visits (Cordeau et al., Networks, 1997)
- Practical Problems
  - Delivery of consumer products to grocery stores
  - Collection of money from vending machines and telephone coin boxes
  - Delivery of heating oil to households

VRP – The Vehicle Routing Problem

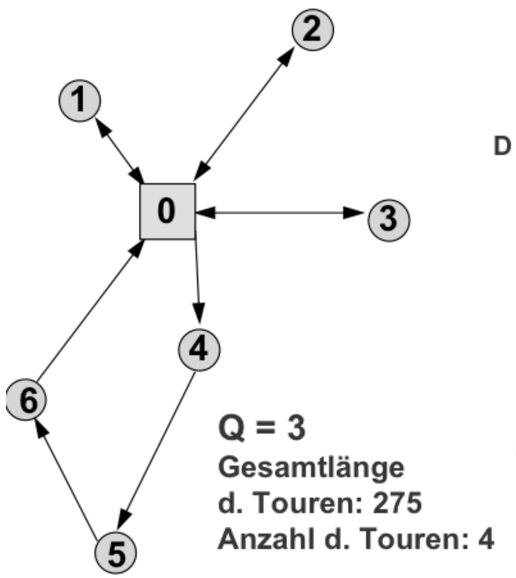
- Structure:
  - Depot
  - m (or at most m) identical vehicles based at the depot
  - n customers
  - Distance (cost, travel time) matrix (cij)
  - q_i : demand of customer i
  - Q: vehicle capacity
  - L: maximal route length (duration)



- VRP – determine a set of m or at most m vehicle routes
  1. Starting and ending at the depot
  2. Visiting each customer exactly once
  3. Satisfying the capacity constraint
  4. Satisfying the maximal length constraint
  5. Of minimal total cost

Heuristics for routing problems

- Classical-Algorithms:
  - Savings (Clarke, Wright, Operations Research, 1965)
  - Sweep (Gillett, Miller, Operations Research, 1974)
  - Cluster first, route second (Fisher, Jaikumas, Net- works, 1981)
  - Intra-route improvement methods (TSP heuristics)
  - Inter-route improvement methods (λ-interchanges, Osman, 1993)
- Metaheuristics
  - Local Search
    - Simulated Annealing
    - Deterministic Annealing
    - Tabu-Search
  - Population Search
    - Genetic Algorithms
  - Learning Mechanisms
    - Neural networks
    - Ant colony systems



D

	0	1	2	3	4	5	6
0	-	20	30	30	20	50	35
1		-	30	45	35	65	45
2			-	30	45	75	55
3				-	35	70	60
4					-	35	25
5						-	25
6							-

S

	1	2	3	4	5	6
1	-	20	5	5	5	10
2		-	30	5	5	10
3			-	15	10	5
4				-	35	30
5					-	60
6						-

SA – Simulated Annealing

- Problemspecific parameters
  - Feasible solution space
  - Neighbours
  - Evaluation
- Generic parameters
  - Temperature
  - Cooling Schedule
  - Termination condition

- Doesn‘t have to return a better point, return an accepted solution
- Acceptance based on the current temperature T
- T updated periodically
- Cooling schedule is key to succes
  - High temperature ➔ High probability of acceptance
  - Cold ➔ Local search
  - Warm phase is effective phase of SA

Osman's SA

DSOR LAB UNIVERSITÄT PADERBORN

- Uses better starting solutions
- Some parameters of the algorithm are adjusted in a trial phase
- Richer solution neighborhoods are explored
- Cooling schedule is more sophisticated
  - Not decreased continuously nor as a step function
  - Decreases continously as long as the current solution is modified
  - Whenever $x_t+1 = x_t$, the temperature is halved or replaced by the temperature at which the incumbent was identified

- Neighborhood structure uses a λ-interchange generation mechanism
  - Two routes p and q are selected,
  - Two subsets of customers S_p and S_q (one from each route)
  - Satisfying $|S_p| \leq \lambda$ and $|S_q| \leq \lambda$
  - Operation swaps the customers of S_p with those of S_q as long as this is feasible

- 2 phases

Osman's SA: Phase 1

- Step 1: initial solution
  - Generate an initial solution
  - By means of Clarke and Wright algorithm (Savings algorithm)

- Step 2: descent
  - Search the solution space using the λ-interchange scheme
  - Implement an improvement as soon as it is identified
  - Stop whenever an entire neighborhood exploration yields no improvement

Osman's SA: Phase 2

- Step 1: initial solution
  - Use solution of Phase 1 or of Clarke and Wright algorithm
  - Perform a complete neighborhood search using the λ-interchange
  - Record Δ_{max} and Δ_{min} ; Compute β
  - $\Theta_1 := \Delta_{max}$, $\delta := 0$, $k := 1$, $k_3 := 3$, $t := 1$, $t^* := 1$
- Step 2: next solution
  - Exploring the neighborhood of xt using λ-interchanges
  - When a solution x.with $f(x) < f(x_t)$ is encountered, set $x_{t+1} := x$
  - If $f(x) < f(x^*)$, set $x^* := x$ and $\Theta^* := \Theta_k$
  - If a whole exploration yields no better solution then set x_{t+1} by probability, $\delta := 1$
- Step3: temperature update
  - Occasional increment rule: if $\delta = 1$, set $\Theta_{t+1} := \max\{\Theta_t/2, \Theta^*\}$, $\delta := 0$ and $k := k+1$
  - Normal decrement rule: if $\delta = 0$, set $\Theta_{t+1} := \Theta_t/[(n\beta+n\sqrt{(t)})\Delta_{max}\ \Delta_{min}]$
  - Stopping criteria: Set $t := t + 1$. If $k = k_3$, stop!
  - Otherwise, go to Step 2

Computational Experiments

- Results on 14 Instances
- Not competative
- Rarelly identifies best solution
- Up to >12% above best known
- Avg. Gap ≈ 2,1%
- There are better heuristics

Instance	f* solution value	Best known	Time	Gap
E051-05e	528,00	524,61	167,40	0,65%
E076-10e	838,62	835,26	6434,30	0,40%
E101-08e	829,80	826,14	9334,00	0,44%
E101-10c	826,00	819,56	632,00	0,79%
E121-07c	1176,00	1042,11	315,80	12,85%
E151-12c	1058,00	1028,42	5012,30	2,88%
E200-17c	1378,00	1291,45	2318,10	6,70%
D051-06c	**555,43**	555,43	3410,20	0,00%
D076-11c	**909,68**	909,68	626,50	0,00%
D101-09c	866,75	865,94	957,20	0,09%
D101-11c	890,00	866,37	305,20	2,73%
D121-11c	1545,98	1541,14	7622,50	0,31%
D151-14c	1164,12	1162,55	84301,20	0,14%
D200-18c	1417,85	1395,85	5708,00	1,58%