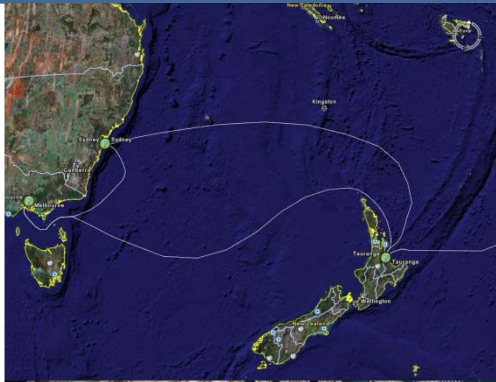
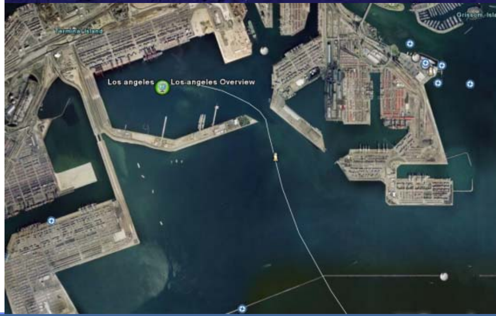
DTU Transport

The Double TSP with Multiple Stacks - Heuristic Solution Approaches



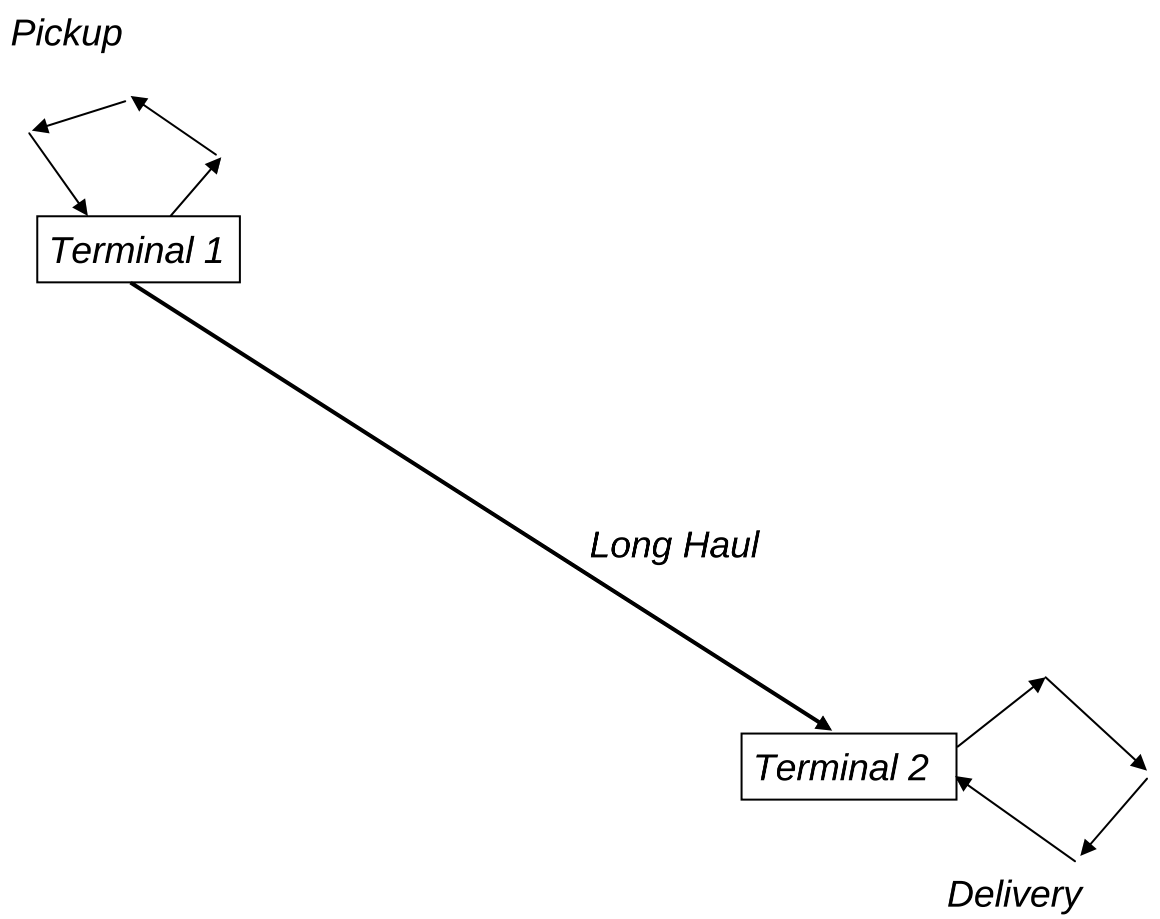


Oli B.G. Madsen
Hanne L. Petersen

Short presentation of DTU Transport

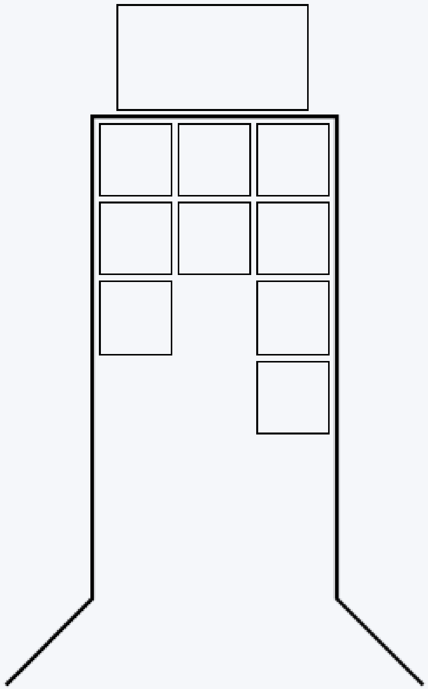
- Danish Transport Research Institute (DTF) and Centre for Traffic and Transport (CTT) merged into one department at DTU on 1st of January 2008
- The new name of the organization is "Department of Transport" or just "DTU Transport"
- One of the 21 centres and departments at DTU
- Teaching, research, industrial collaboration and public sector consultancy
- Around 65-70 employees (3 full professors)
- 22 courses (2 shared courses)
- More than 68% externally financed
- 60 external projects
- Formal Collaboration with 4 Leading Universities
- Informal Collaboration with 10 Leading Universities

The Double TSP with Multiple Stacks - example



Problem Definition DTSPMS

- all orders served by the same container, no repacking
- all items uniform, no stacking
- sequencing constraints on loading, the available loading positions form a grid on the floor of the container



Usually 3x11=33 boxes in a 40-foot container

Demonstration

Problem Definition

- Pickup and delivery problem from real life
    - pickups and deliveries are performed in separated areas/graphs, ie. all pickups lie before all deliveries
    - each graph has a depot, and the transport between the two depots is not considered
    - all orders served by the same container, no repacking
    - each order consists of one item, which has a pickup address and a delivery address
    - all items uniform, no stacking
    - sequencing constraints on loading, the available loading positions form a grid on the floor of the container

Description of the problem

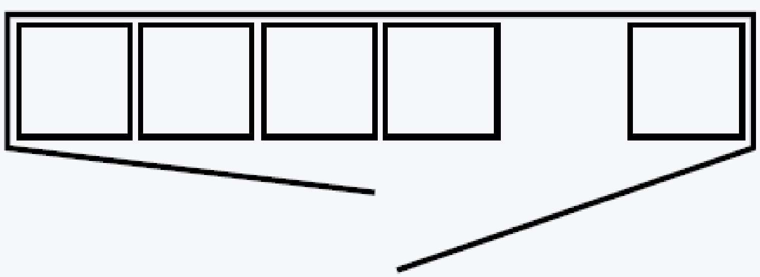
- Two depots (terminals) – one for pickup – one for delivery
- Long distance between the 2 depots
- One pickup route and one delivery route
- One order per customer (rectangular boxes e.g. Euro pallets)
- Each order has an origin and a destination
- All boxes have the same size
- The boxes are placed in a container in a given number of rows (horizontal stacks)
- The rows are mutually independent
- The container is accessed from the opening in one end of the container
- The container can not be repacked during the complete journey meaning the loading in each row is subject to a LIFO constraint
- No time windows are considered

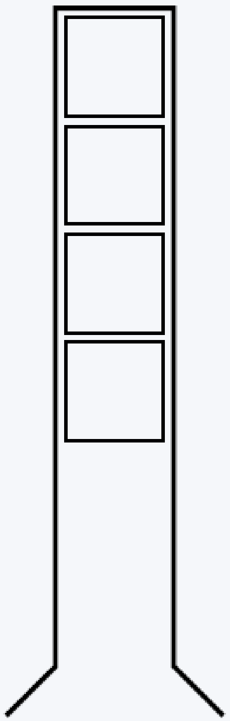
The Problem

- It is an extension of the regular travelling salesman problem (TSP), with
  - pickups and deliveries
  - multiple loading rows (individually accessible)
- Given: A set of orders, each with a pickup and a delivery address.
- Produce:
  - pickup route
  - delivery route
  - loading plan
- Objective: Minimise total travelled distance

Special Cases

- only one loading row → pickup and delivery routes must be exact opposites (UB)
  - solve a TSP on the sum of the distance matrices
- n rows → the two routes are completely independent (LB)
  - solve two TSPs





A Mathematical Model

First variable: $x_{ij}^G = \begin{cases} 1, \text{ if edge } (i,j) \text{ is used in graph } G \\ 0, \text{ otherwise} \end{cases}$

Objective:

Symmetric edge costs

$$\min \sum_{i,j \in N_0, G \in \{P,D\}} c_{ij}^G \cdot x_{ij}^G$$

Flow balance:

$$\sum_i x_{ij}^G = 1 \qquad \forall j \in N_0$$

$$\sum_j x_{ij}^G = 1 \qquad \forall i \in N_0$$

Precedence Constraints 1

Second variable: $y_{ij}^G = \begin{cases} 1, \text{ if } i \text{ is visited before } j \text{ in graph } G \\ 0, \text{ otherwise} \end{cases}$

$$x_{ij}^G \leq y_{ij}^G \qquad \forall i, j, G$$

$$y_{ik}^G + y_{kj}^G \leq y_{ij}^G + 1 \qquad \forall i, j, k, G$$

$$y_{ij}^G + y_{ji}^G = 1 \qquad \forall i, j, G, i \neq j$$

- implicit subtour elimination

Precedence Constraints 2

Precedences are only relevant when two items in the same row.

Third variable: $z_{ir} = \begin{cases} 1, \text{ if } i \text{ is placed in row } r \\ 0, \text{ otherwise} \end{cases}$

$$y^{\mathrm{P}}_{ij} + z_{ir} + z_{jr} \leq 2 + y^{\mathrm{D}}_{ji} \qquad \forall i, j, r$$

Finally, keep track of the row assignments:

$$\sum_r z_{ir} = 1 \qquad \forall i$$

$$\sum_i z_{ir} \leq L \qquad \forall r$$

Mathematical Model - Constraints

$$\sum_i x_{ij}^G = 1 \qquad \forall j \in V^G \qquad (2)$$

$$\sum_j x_{ij}^G = 1 \qquad \forall i \in V^G \qquad (3)$$

$$y_{ij}^G + y_{ji}^G = 1 \qquad \forall i, j, G, i \neq j \qquad (4)$$

$$y_{ik}^G + y_{kj}^G \leq y_{ij}^G + 1 \qquad \forall i, j, k, G \qquad (5)$$

$$x_{ij}^G \leq y_{ij}^G \qquad \forall i, j, G \qquad (6)$$

$$y_{ij}^{\mathrm{P}} + z_{ir} + z_{jr} \leq 3 - y_{ij}^{\mathrm{D}} \qquad \forall i, j, r = 1, \ldots, R \qquad (7)$$

$$\sum_r z_{ir} = 1 \qquad \forall i \qquad (8)$$

$$\sum_i z_{ir} = L \qquad \forall r = 1, \ldots, R \qquad (9)$$

$$x, y, z \in \mathbb{B} \qquad (10)$$

- flow conservation constraints

$$\sum_i x_{ij}^G = 1$$

$$\sum_j x_{ij}^G = 1$$

- setting a precedence variable

$$y_{ij}^G + y_{ji}^G = 1$$

- i before k and k before j means i before j

$$y_{ik}^G + y_{kj}^G \leq y_{ij}^G + 1$$

- if edge (i,j) is used the the corresponding precedence variable is set

$$x_{ij}^G \leq y_{ij}^G$$

- LIFO constraint if i and j in same row

$$y_{ij}^{\mathrm{P}} + z_{ir} + z_{jr} \leq 3 - y_{ij}^{\mathrm{D}}$$

- assignment constraint

$$\sum_r z_{ir} = 1$$

- max length of a row

$$\sum_i z_{ir} = L$$

- x, y, z binary

$$x, y, z \in \mathbb{B}$$

Size of the Model

$$\sum_i x_{ij}^G = 1$$
$$\sum_j x_{ij}^G = 1$$
$$y_{ij}^G + y_{ji}^G = 1$$
$$y_{ik}^G + y_{kj}^G \leq y_{ij}^G + 1$$
$$x_{ij}^G \leq y_{ij}^G$$
$$y_{ij}^{\mathrm{P}} + z_{ir} + z_{jr} \leq 3 - y_{ij}^{\mathrm{D}}$$
$$\sum_r z_{ir} = 1$$
$$\sum_i z_{ir} = L$$
$$x, y, z \in \mathbb{B}$$

- Typically 3 rows each containing 11 boxes
- In total 78.606 constraints
- Constraints (4): 71.874 constraints
- 4.455 variables

- CPLEX can only solve 2 by 5 or 3 by 4 to optimallity within an hour of CPU time

Alternative Models

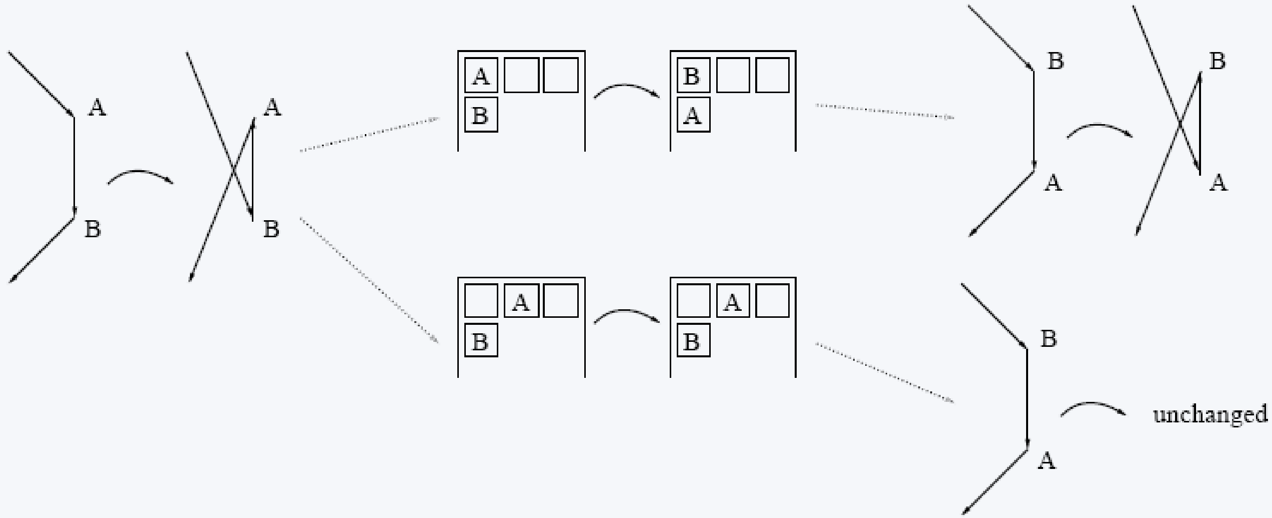
Five other models has been analyzed

Heuristic Approaches

- Tabu Search (TS)
- Simulated Annealing (SA)
- Steepest Descent (SD)
- Large Neighbourhood Search (LNS)

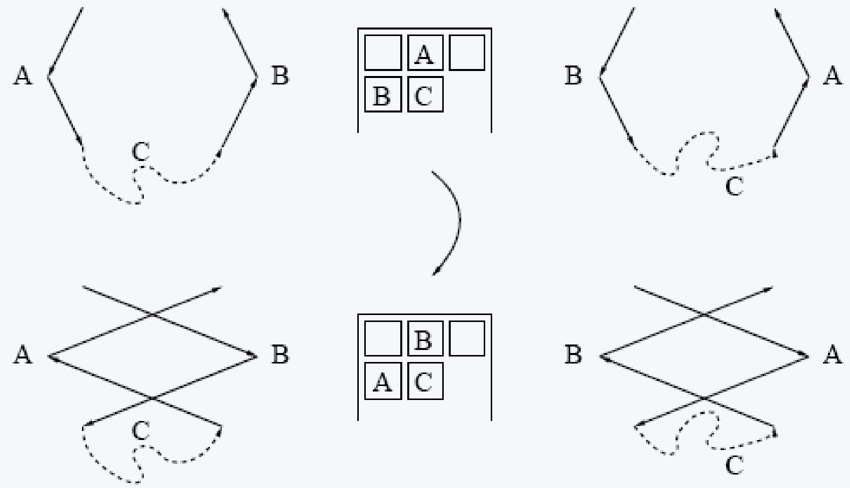
Two Different Move Structures (TS, SD and SA)

- Change routing, keep row assignment
  - swap any pair of customers that are adjacent in either route
  - if they are in the same row, also swap in opposite route (loading positions will be swapped)
  - if they are in separate rows, nothing else needs to be changed (loading positions will be unchanged)



Two Different Move Structures (TS, SD and SA)

- Swap rows
  - swap the loading positions of any two items that are in separate rows
  - also swap their positions in both routes



- In the end both moves are necessary to cover the solution space

LNS

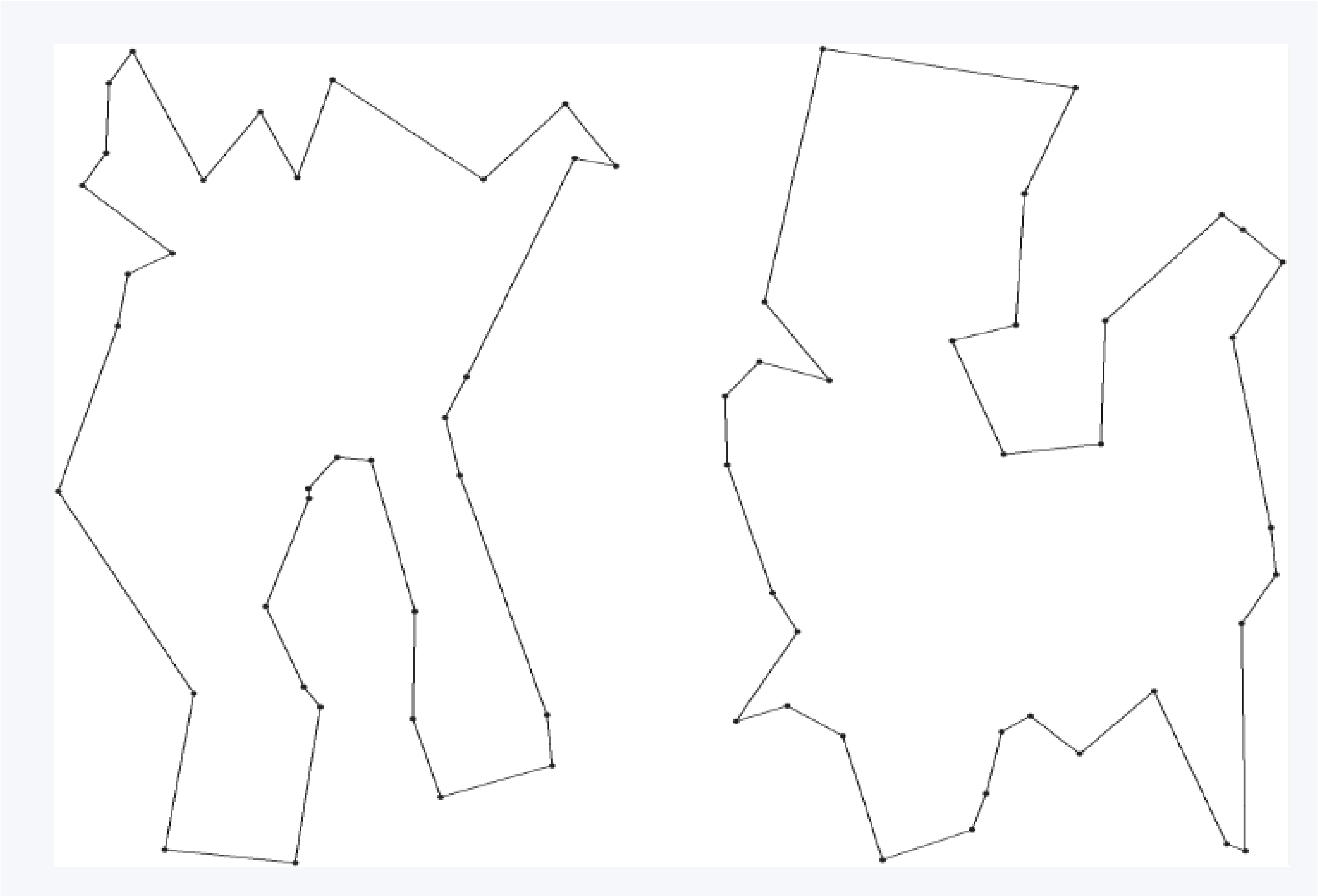
- Two removal strategies
  - Remove **orders** similar to those already removed
  - Remove orders that are the most expensive to cover in the current solution
- Reinsertion based on
  - Nearest
  - Farthest
  - Cheapest
  - Most expensive
  - Plus noise to increase diversification

Initial Feasible Solutions

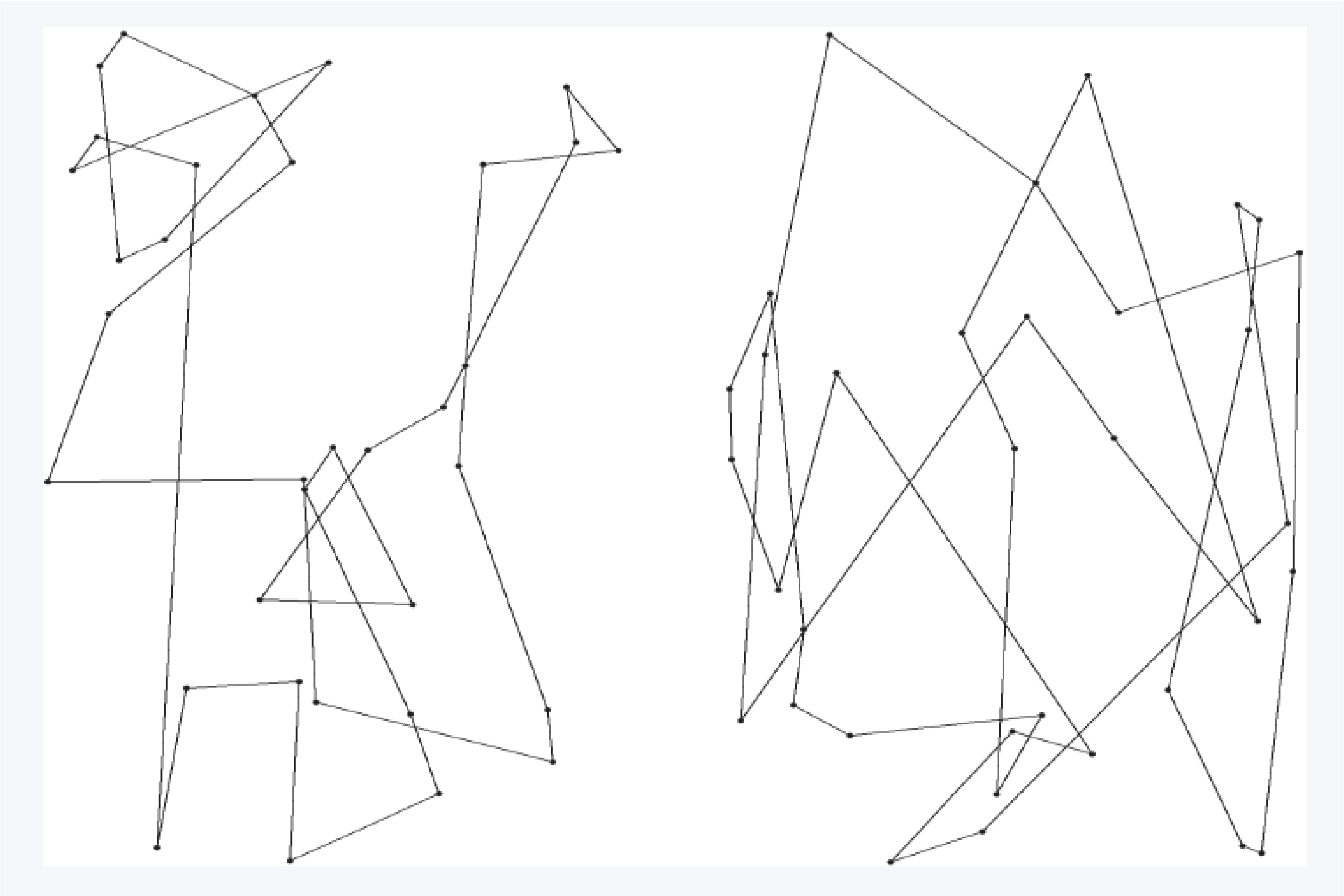
- For TS, SA and LNS
  - Solve the single stack problem
  - By adding the two distance matrices
  - And solving a regular TSP by e.g. Savings

- For SD
  - A number of initial solutions by randomly generating an ordering for the pickup route and reversing this for the delivery route

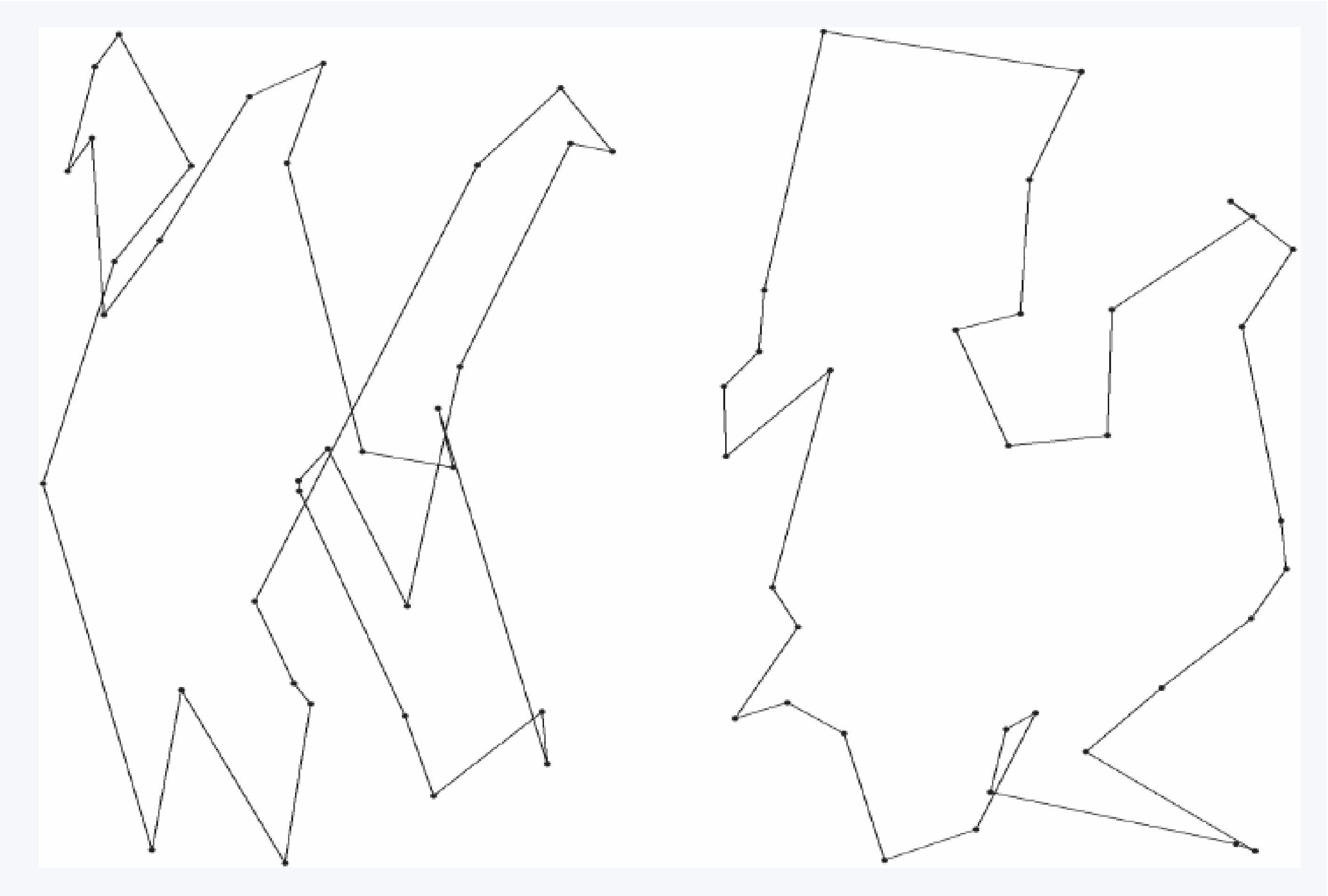
TSP Solution – 33 Customers



Initial Solution – Savings Algorithm



Best Heuristic Solution



Test Instances

- Randomly generated in two 100 x 100 squares with the terminals located in the middle of the squares
- Euclidean distances rounded to the nearest integer
- It may violate the triangle inequality (The present solution methods do not rely on this)
- Two test sets each containing ten problems of size 33 orders (3 x 11), i.e. 33 customers in each graph
- The first set is used for parameter tuning
- The second set is used for testing
- One-line run (10 seconds) and reasonable waiting (3 minutes)
- To find best solutions: 1 hour CPU
- Test instances on http://www.imm.dtu.dk/~hlp

Lower and Upper Bounds

- nS: lower bound (n-stack problem)
- LP: LP relaxation
- root: root node after CPLEX cuts
- BB: best bound after 1 hour CPLEX
- Best: Best known feasible solution obtained by using 1 hour CPU time
- qual.: Best/nS
- SS: optimal solution to the single stack problem
- init: heuristic solution to the single stack problem

Table 1: Bounds

	nS	LP	root	BB	Best	qual.	init	SS
R00	911	793	802	812	1063	1.17	1886	1685
R01	875	820	824	831	1032	1.18	1690	1581
R02	935	826	844	845	1065	1.14	1672	1563
R03	961	887	905	905	1100	1.14	1836	1745
R04	937	859	860	863	1052	1.12	1671	1628
R05	900	811	814	816	1008	1.12	1548	1439
R06	998	941	944	944	1110	1.11	1739	1644
R07	963	894	900	900	1105	1.15	1867	1695
R08	978	899	911	922	1109	1.13	1761	1636
R09	976	889	909	910	1092	1.12	1610	1553
R10	901	822	833	839	1016	1.13	1697	1575
R11	892	810	820	823	1001	1.12	1494	1429
R12	984	934	946	950	1109	1.13	1778	1673
R13	956	887	895	897	1085	1.13	1707	1613
R14	879	794	803	803	1034	1.18	1704	1565
R15	985	903	916	917	1142	1.16	1943	1783
R16	967	857	887	894	1094	1.13	1767	1647
R17	946	847	882	884	1073	1.13	1716	1620
R18	1008	876	920	921	1126	1.12	1796	1673
R19	938	839	855	864	1097	1.17	1725	1633

Lower and Upper Bounds

- nS: lower bound (n-stack problem)
- LP: LP relaxation
- root: root node after CPLEX cuts
- BB: best bound after 1 hour CPLEX
- Best: Best known feasible solution obtained by using 1 hour CPU time
- qual.: Best/nS
- SS: optimal solution to the single stack problem
- init: heuristic solution to the single stack problem

- n-stack best lower bound (still weak)
- Suggests that Best is rather good

Table 1: Bounds

	nS	LP	root	BB	Best	qual.	init	SS
R00	911	793	802	812	1063	1.17	1886	1685
R01	875	820	824	831	1032	1.18	1690	1581
R02	935	826	844	845	1065	1.14	1672	1563
R03	961	887	905	905	1100	1.14	1836	1745
R04	937	859	860	863	1052	1.12	1671	1628
R05	900	811	814	816	1008	1.12	1548	1439
R06	998	941	944	944	1110	1.11	1739	1644
R07	963	894	900	900	1105	1.15	1867	1695
R08	978	899	911	922	1109	1.13	1761	1636
R09	976	889	909	910	1092	1.12	1610	1553
R10	901	822	833	839	1016	1.13	1697	1575
R11	892	810	820	823	1001	1.12	1494	1429
R12	984	934	946	950	1109	1.13	1778	1673
R13	956	887	895	897	1085	1.13	1707	1613
R14	879	794	803	803	1034	1.18	1704	1565
R15	985	903	916	917	1142	1.16	1943	1783
R16	967	857	887	894	1094	1.13	1767	1647
R17	946	847	882	884	1073	1.13	1716	1620
R18	1008	876	920	921	1126	1.12	1796	1673
R19	938	839	855	864	1097	1.17	1725	1633

Results - Summary

The solution quality =

the objective value of the solution to an actual instance divided by

the objective value of the best known solution for that instance

LNS: Large Neighbourhood Search

SA: Simulated Annealing

TS: Tabu Search

SD: Steepest Descent

	10 seconds				3 minutes			
	LNS	SA	TS	SD	LNS	SA	TS	SD
R00	1.04	1.26	1.42	1.66	1.01	1.13	1.23	1.58
R01	1.04	1.17	1.34	1.64	1.01	1.08	1.21	1.61
R02	1.04	1.19	1.38	1.57	1.01	1.10	1.24	1.53
R03	1.06	1.22	1.44	1.66	1.01	1.12	1.24	1.62
R04	1.05	1.25	1.38	1.59	1.02	1.11	1.23	1.56
R05	1.03	1.22	1.25	1.53	1.01	1.15	1.21	1.47
R06	1.06	1.19	1.37	1.56	1.02	1.11	1.26	1.51
R07	1.05	1.23	1.39	1.66	1.01	1.11	1.26	1.58
R08	1.04	1.21	1.36	1.56	1.01	1.11	1.27	1.51
R09	1.04	1.15	1.29	1.47	1.01	1.08	1.19	1.46
R10	1.05	1.24	1.45	1.67	1.00	1.15	1.25	1.64
R11	1.06	1.24	1.24	1.49	1.01	1.10	1.22	1.48
R12	1.04	1.21	1.44	1.60	1.01	1.13	1.22	1.55
R13	1.04	1.23	1.37	1.55	1.01	1.08	1.22	1.53
R14	1.03	1.22	1.47	1.63	1.00	1.11	1.25	1.58
R15	1.04	1.21	1.37	1.62	1.01	1.10	1.26	1.56
R16	1.02	1.20	1.35	1.61	1.00	1.10	1.18	1.55
R17	1.04	1.24	1.48	1.60	1.00	1.12	1.28	1.58
R18	1.05	1.20	1.33	1.59	1.01	1.13	1.21	1.53
R19	1.03	1.16	1.31	1.57	1.01	1.11	1.25	1.54
Avg. set 0	1.04	1.21	1.36	1.59	1.01	1.11	1.23	1.54
Avg. set 1	1.04	1.22	1.38	1.59	1.01	1.11	1.23	1.56

Results - Summary

The solution quality =

the objective value of the solution to an actual instance divided by

the objective value of the best known solution for that instance

LNS: Large Neighbourhood Search

SA: Simulated Annealing

TS: Tabu Search

SD: Steepest Descent

- LNS consistently shows best results
- Increasing running time improves the solution (apart from SD)

	10 seconds				3 minutes			
	LNS	SA	TS	SD	LNS	SA	TS	SD
R00	1.04	1.26	1.42	1.66	1.01	1.13	1.23	1.58
R01	1.04	1.17	1.34	1.64	1.01	1.08	1.21	1.61
R02	1.04	1.19	1.38	1.57	1.01	1.10	1.24	1.53
R03	1.06	1.22	1.44	1.66	1.01	1.12	1.24	1.62
R04	1.05	1.25	1.38	1.59	1.02	1.11	1.23	1.56
R05	1.03	1.22	1.25	1.53	1.01	1.15	1.21	1.47
R06	1.06	1.19	1.37	1.56	1.02	1.11	1.26	1.51
R07	1.05	1.23	1.39	1.66	1.01	1.11	1.26	1.58
R08	1.04	1.21	1.36	1.56	1.01	1.11	1.27	1.51
R09	1.04	1.15	1.29	1.47	1.01	1.08	1.19	1.46
R10	1.05	1.24	1.45	1.67	1.00	1.15	1.25	1.64
R11	1.06	1.24	1.24	1.49	1.01	1.10	1.22	1.48
R12	1.04	1.21	1.44	1.60	1.01	1.13	1.22	1.55
R13	1.04	1.23	1.37	1.55	1.01	1.08	1.22	1.53
R14	1.03	1.22	1.47	1.63	1.00	1.11	1.25	1.58
R15	1.04	1.21	1.37	1.62	1.01	1.10	1.26	1.56
R16	1.02	1.20	1.35	1.61	1.00	1.10	1.18	1.55
R17	1.04	1.24	1.48	1.60	1.00	1.12	1.28	1.58
R18	1.05	1.20	1.33	1.59	1.01	1.13	1.21	1.53
R19	1.03	1.16	1.31	1.57	1.01	1.11	1.25	1.54
Avg. set 0	1.04	1.21	1.36	1.59	1.01	1.11	1.23	1.54
Avg. set 1	1.04	1.22	1.38	1.59	1.01	1.11	1.23	1.56

Instances with Known Optimal Solutions

- Optimal 12 customer solutions by CPLEX
- LSN found the optimal solution in all test instances (60 out of 60)
- SA found the optimal solution in 49 out of 60 test instances (Gap < 2%)
- TS found the optimal solution in 0 out of 60 test instances (Gap < 8%)

Conclusions and Future Work

- Conclusions
  - A new variant of the TSP with pickup and delivery is introduced
  - Four metaheuristic approaches have been implemented
  - Large Neighbourhood Search best with 3-minute solutions within 2% of the best known solutions

- Future Work
  - Improving the heuristic solution methods
  - Improving the optimal methods by relaxing the most numerous constraint
  - Generalize the problem to deal with
    - Several vehicles/containers
    - Several terminals

Thank you for your attention

QUESTIONS

Contact address:
Oli B.G. Madsen, professor, dr.techn.
DTU Transport
DTU, Bygning 115, Bygningstorvet
DK 2800 Kgs. Lungby
(+45) 4525 1526
ogm@transport.dtu.dk
www.transport.dtu.dk