Application oriented vehicle problems in public bus transportation

Gábor Galambos
University of Szeged, Hungary

Joint work with

Viktor Árgilán, János Balogh, József Békési, Balázs Dávid, Miklós Krész, Attila Tóth

Workshop on Traffic Optimization
Heidelberg

8.october 2015.

Szeged

The city of Szeged

Szeged
The University of Szeged

Szeged

Public transportation in Szeged

Motivation

- R&D Project

Overview

- Scheduling problems in public transportation
- Vehicle scheduling
- Vehicle assignment
- „Driver-friendly” vehicle schedules
- Vehicle rescheduling

Scheduling problems in public transportation

For public transportation services certain number of stations, and previously determined – bus or other vehicles – lines are given. Each line connects a pair of stations.

The lines are fixed in timetables which provides the departure and arrival time of the trips for each line, and – sometimes – further services for each day are also fixed.

One of the most important subject of a public transport company is to minimize its operational costs.

Scheduling problems in public transportation (Operations planning)

Routing

↓

Line frequencies

↓

Timetable

↓

Scheduling

„Outside controled”

Scheduling problems in public transportation

This complex problem is divided into subtasks and they are to be solved as separated optimization problems

Vehicle scheduling

↓

Driver scheduling

↓

Driver rostering

Vehicle Scheduling Problem

Vehicle scheduling problem

Input:

- Timetabled trips
- Deadhead trips
- Depots
  - Bus types (solo, double, etc.)
  - Locations

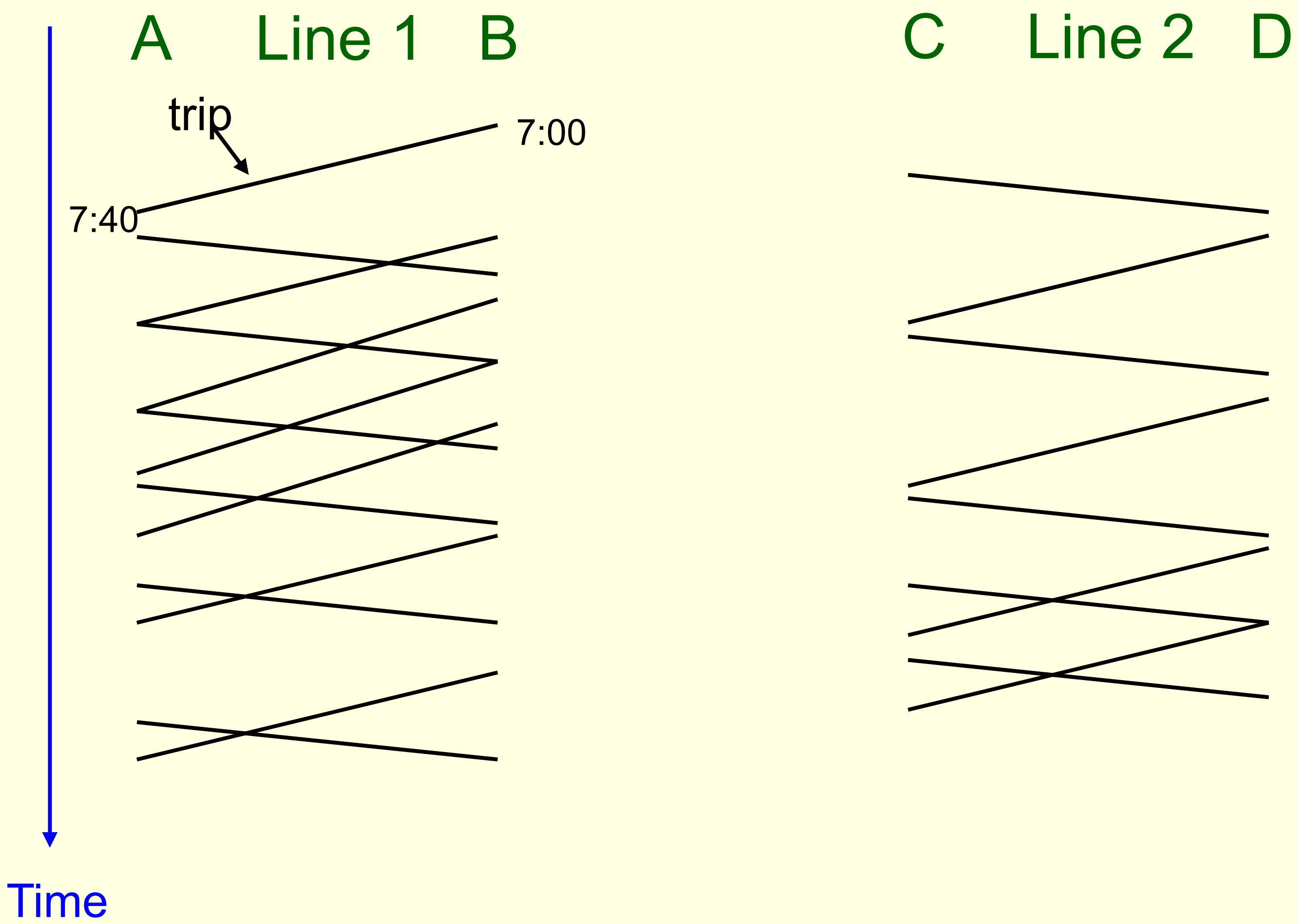
Costs:

- Daily cost of a vehicle (maintanance)
- Cost of the covered distance (transportation costs, deadhead costs)

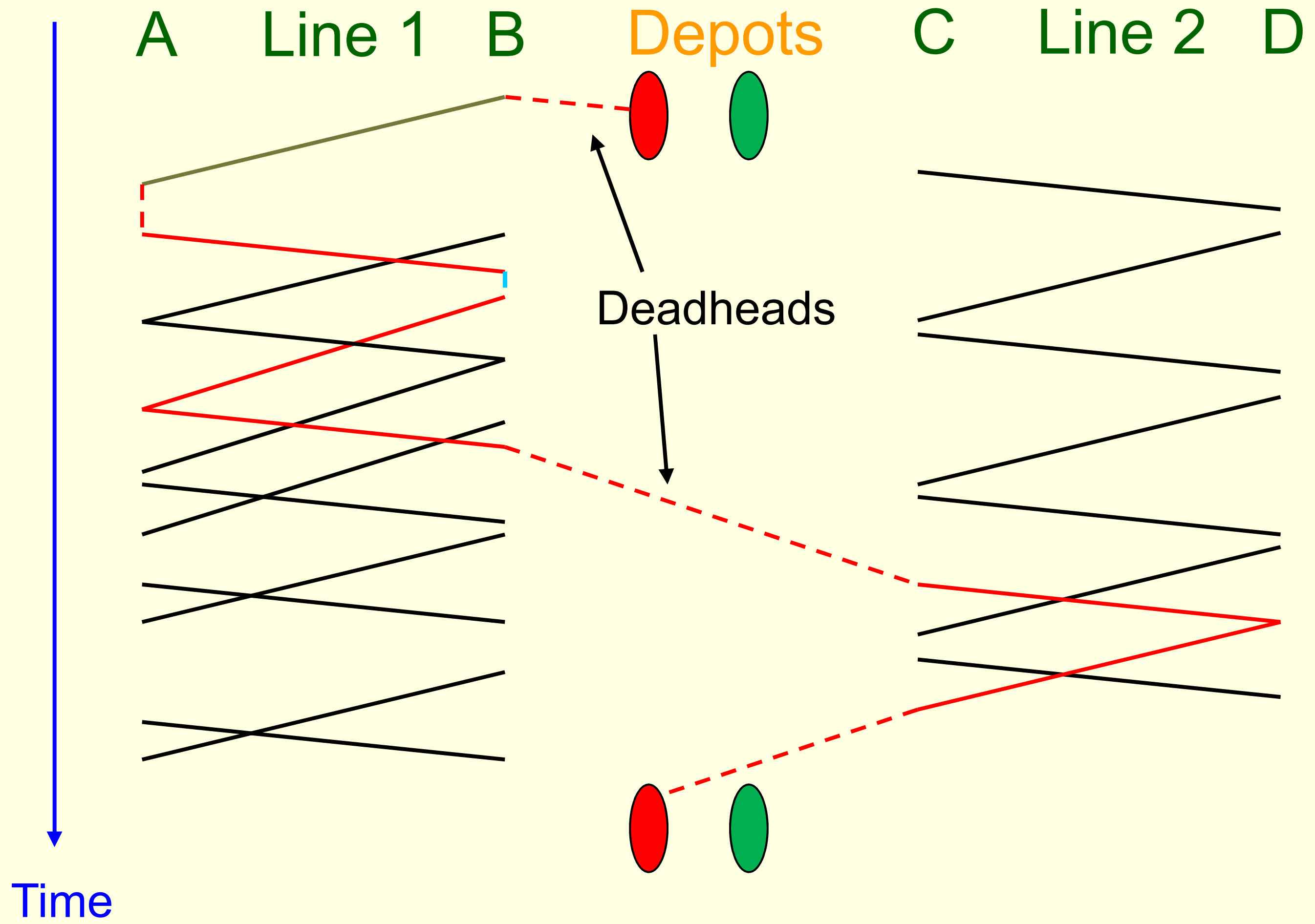
Aim:

- Execute each trip exactly once (with regards to depot compatibility and capacities), minimizing the cost

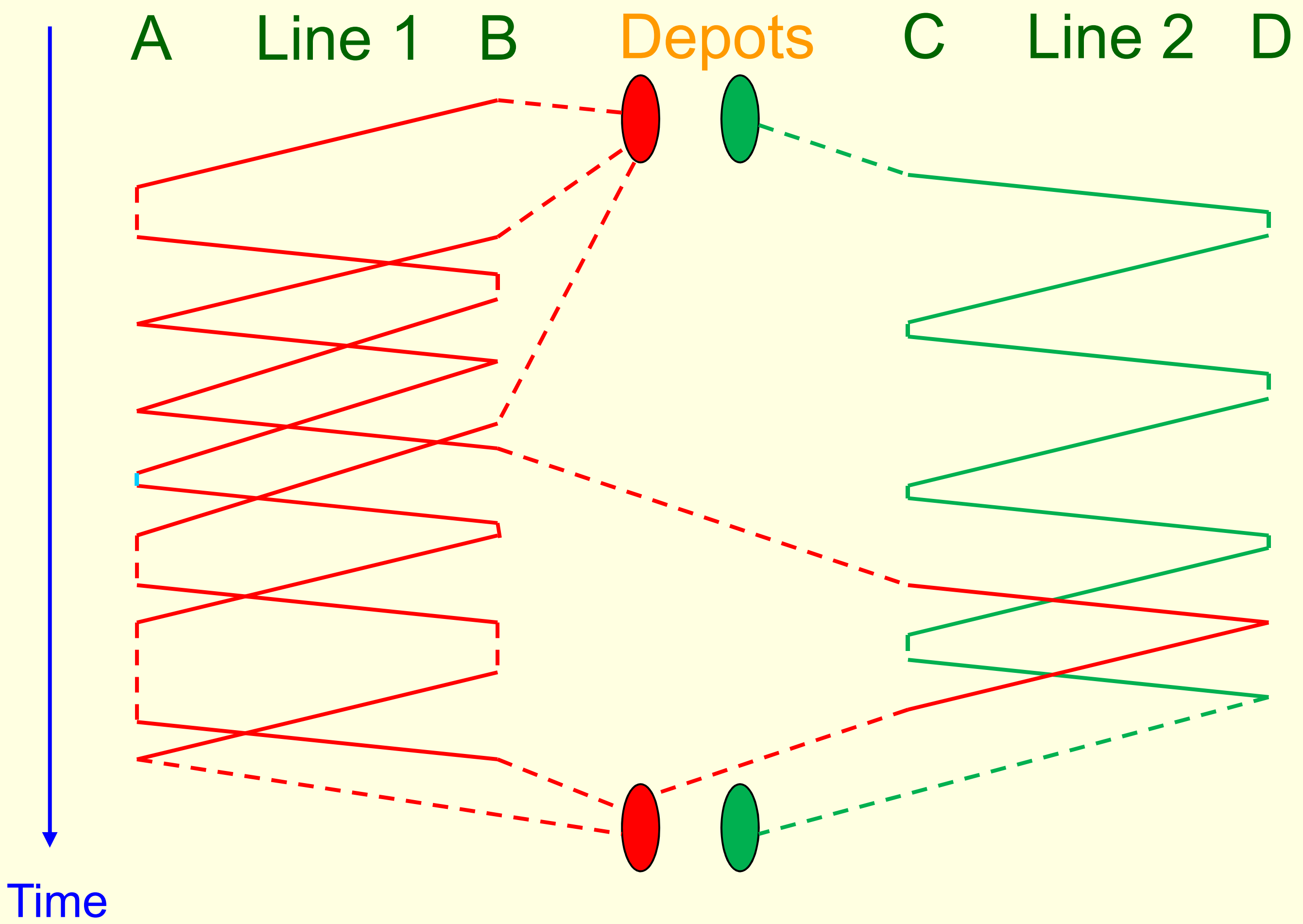
Vehicle scheduling problem



Vehicle scheduling problem



Vehicle scheduling problem

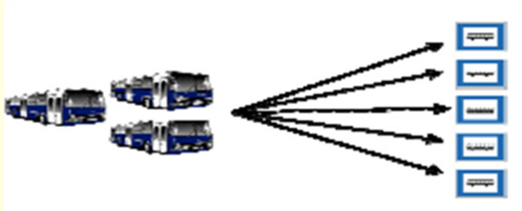


Vehicle scheduling problem

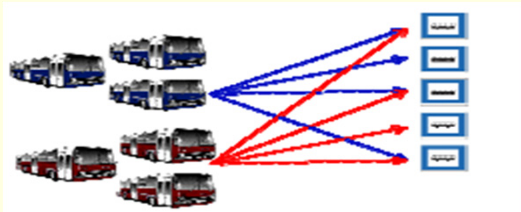
- Single-Depot Vehicle Scheduling Problem(SDVSP)
  - Solvable in polynomial time

  (Matching problem,

  Minimum cost network flow)



- Multiple-Depot Vehicle Scheduling Problem(MDVSP)
  - NP-hard (Bertossi et al., 1987)



Solution by multi-commodity network flow minimization

- Connection based network model (Löbel, 1997)
- Time space network model (Kliewer et al., 2006)

Connection based network

D = set of depots

U = set of trips

V = set of vehicles

$D_u \subseteq D$ set of those depots which can serve the trip $u \in U$

$U_d \subseteq U$ set of those trips which can be served by the depot $d \in D$

$at(d)$ = arrival time to the depot d

$dt(d)$ = departure time from the depot d

$at(u)$ = arrival time of the trip u

$dt(u)$ = departure time of the trip u

Trips u_i and u_j are compatible if $at(u_i) \leq dt(u_j)$ and $dt(u_j) - at(u_i) \geq T_{ij}$

Connection based network
(define a network)

Let N be the set of vertices of the network. Then

$$N = \{\, dt(u) |\, u \in U\} \,\cup \{\, at(u) |\, u \in U\} \cup \{\, dt(d) |\, d \in D\} \cup \{\, at(d) |\, d \in D\}$$

If E_d is the set of timetabled trips belonging to the depot d, then

$$E_d = \{(dt(u), at(u)) |\, u \in U_d\}, \qquad \forall d \in D$$
$$B_d = \{(at(u), dt(v)) |\, u, v \in U_d \text{ are compatible}\}, \forall d \in D$$

where B_d is the set of the depot deadhead trips.

Connection based network

(define a network)

Let R_d be the set of ingoing and outgoing trips of depot d, then

$$R_d = \{ \left(dt(d), dt(u)\right), \left(at(u), at(d)\right) \middle| \, u \in U_d\}, \qquad \forall d \in D$$

Let K_d is the set of depot-circle arcs of the depot d, then

$$K_d = \{(at(d), dt(d)\}, \qquad \forall d \in D$$

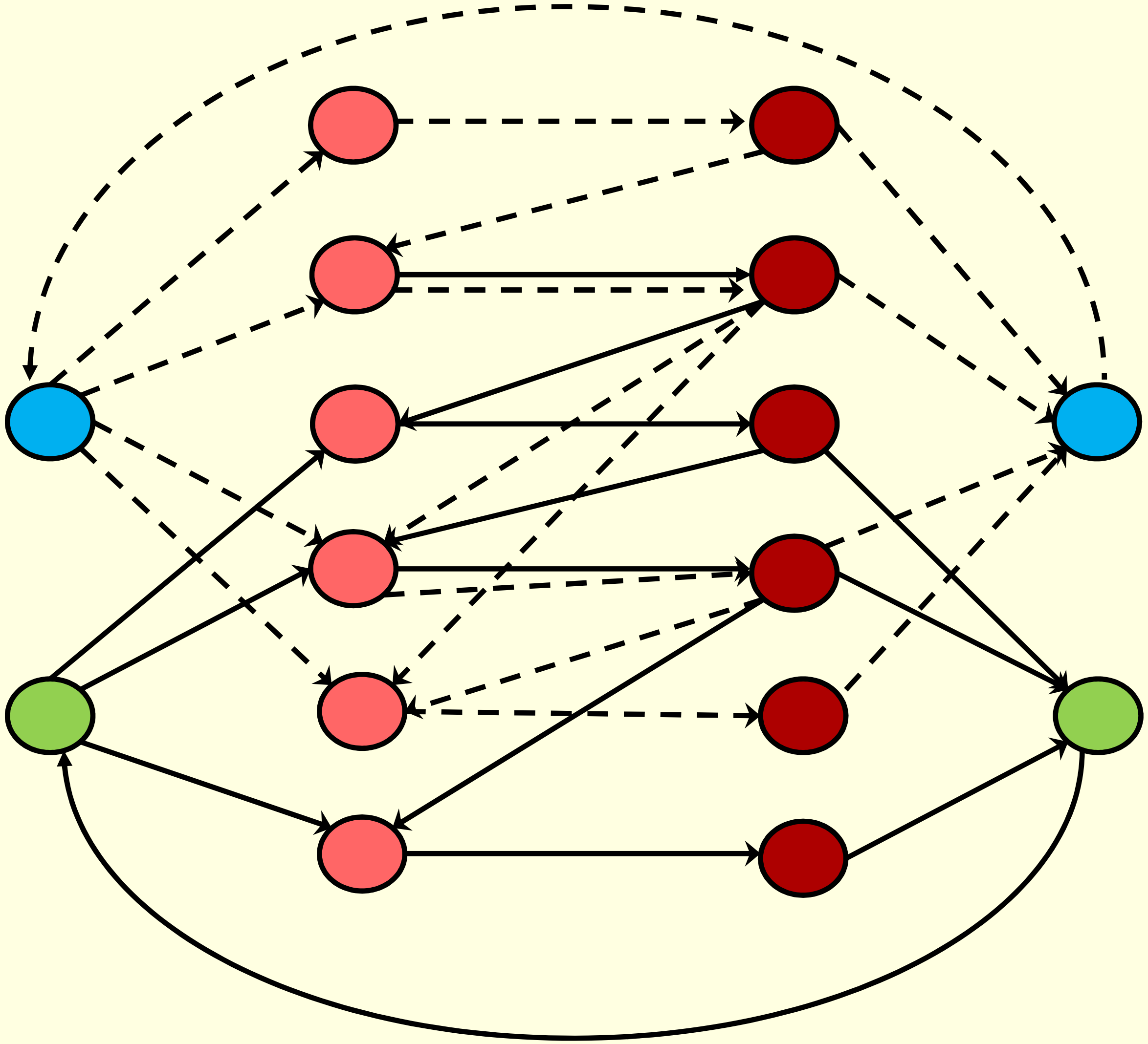
Then the set of arcs belonging to the depot d in the network is

$$A_d = E_d \cup B_d \cup R_d \cup K_d$$

and

$$E = \cup_{d \in D} A_d$$

Connection based network
(define a network)



Connection based network

(define MDVSP on the network)

So, we have a directed graph $G = (N,E)$. We can define the MDVSP on this graph as follows:

Let X be an integer vector with dimension of $|E|$. Its component is belonging to the arc $e \in E$ is x_e^d if $e \in A_d$.
X represent a multicommodity flow.

Considering a schedule S. Then x_e^d we can state

$$x_e^d = \begin{cases} 1, & if \ \ x_e^d \in S \\ 0, & otherwise \end{cases}$$

Connection based network
(define MDVSP on the network)

1. No common edges for two trips:

$$\sum_{d\epsilon Du,\ e=(dt(u),at(u))\epsilon Ed} x_e^d = 1\ \ \forall u\ \epsilon\ U.$$

2. Vehicles must return to the depots:

$$\sum_{e\epsilon n+} x_e^d - \sum_{e\epsilon n-} x_e^d = 0, \forall u\ \epsilon\ U, \forall n\ \epsilon\ N$$

3. If we have capacity constraint for the depots:

$$x_{at(d),dt(d)}^d \leq k_d$$

Connection based network
(define MDVSP on the network)

The objective function is

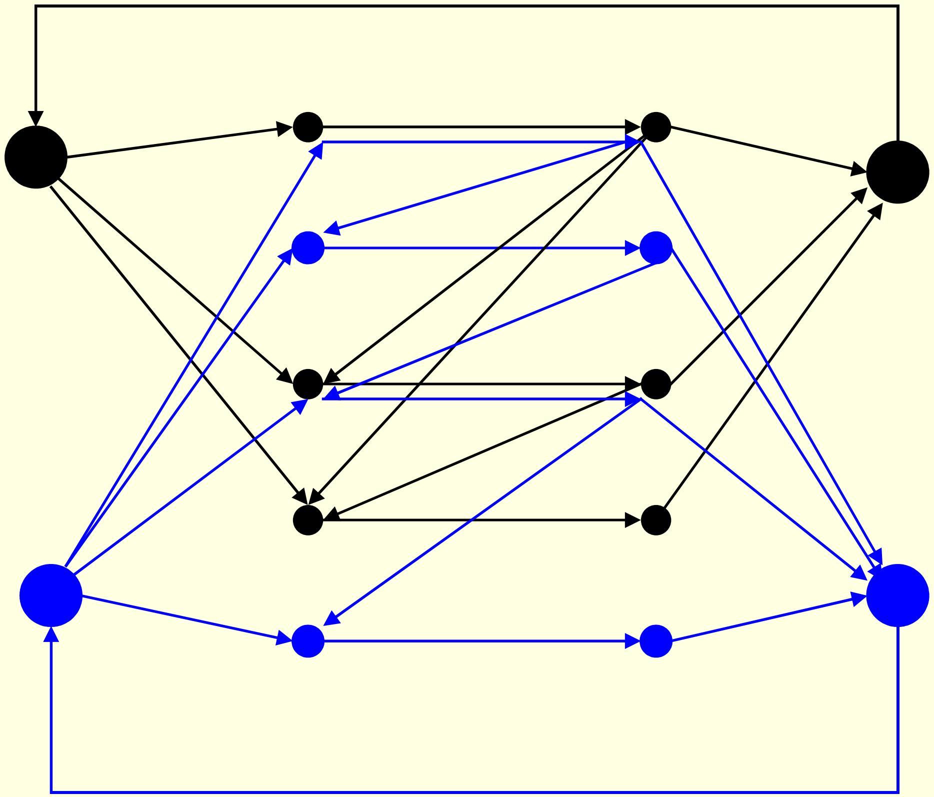
$$min \sum_{e \epsilon E} c_e x_e$$

where c_e is the cost of the edge e.

Connection based network
(define MDVSP on the network)

- Large size
  - Due to high number of possible deadheads

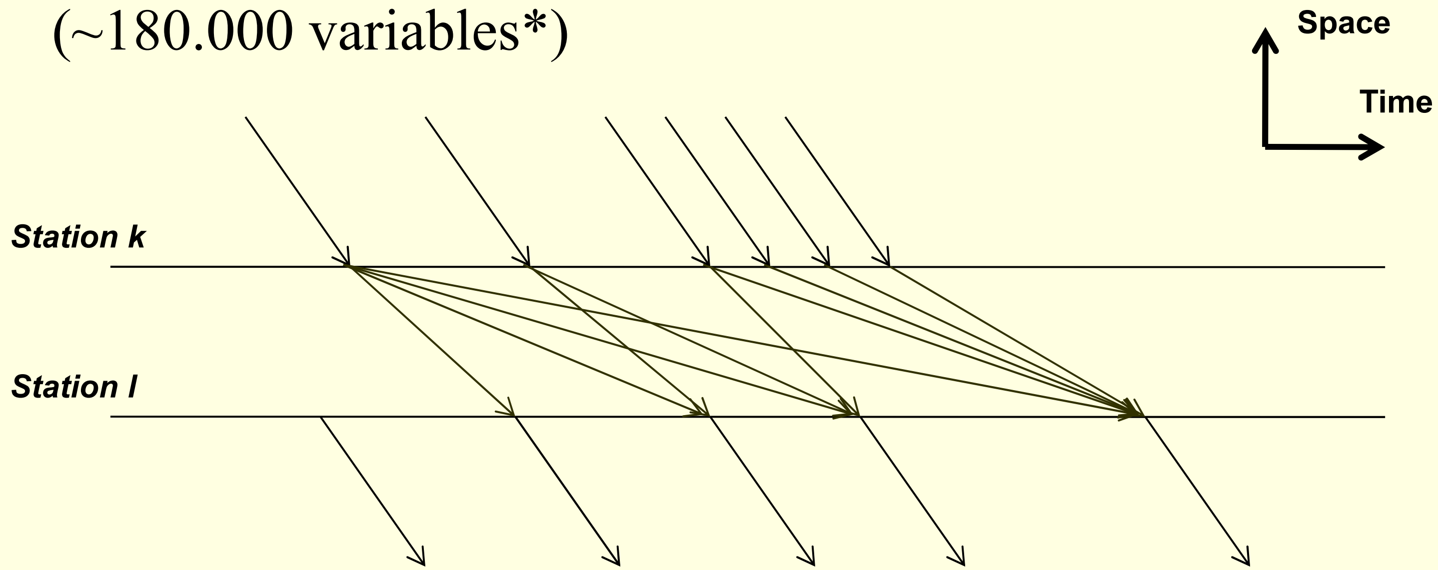
(~5.000.000 variables*)



*Tisza Volán test case

Time space network

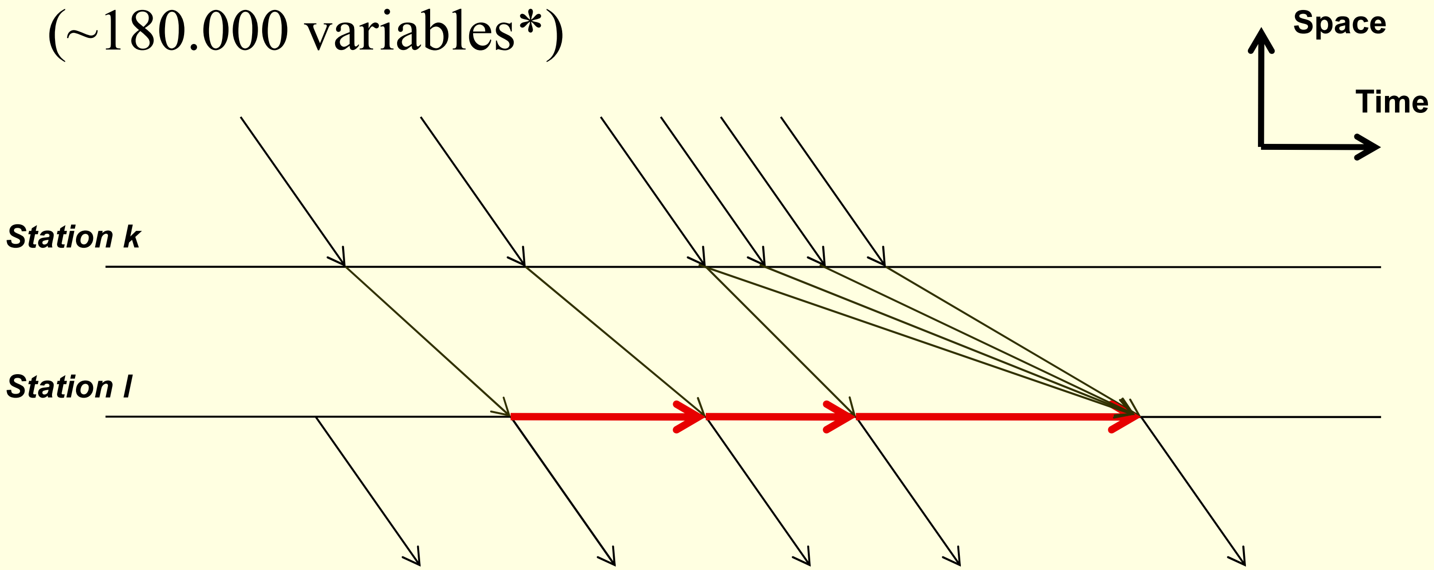
- Introduces timelines
- Nodes are points of time
- Aggregates possible connections into waiting edges
  - Significant decrease in model size (~180.000 variables*)



*Tisza Volán test case

Time space network

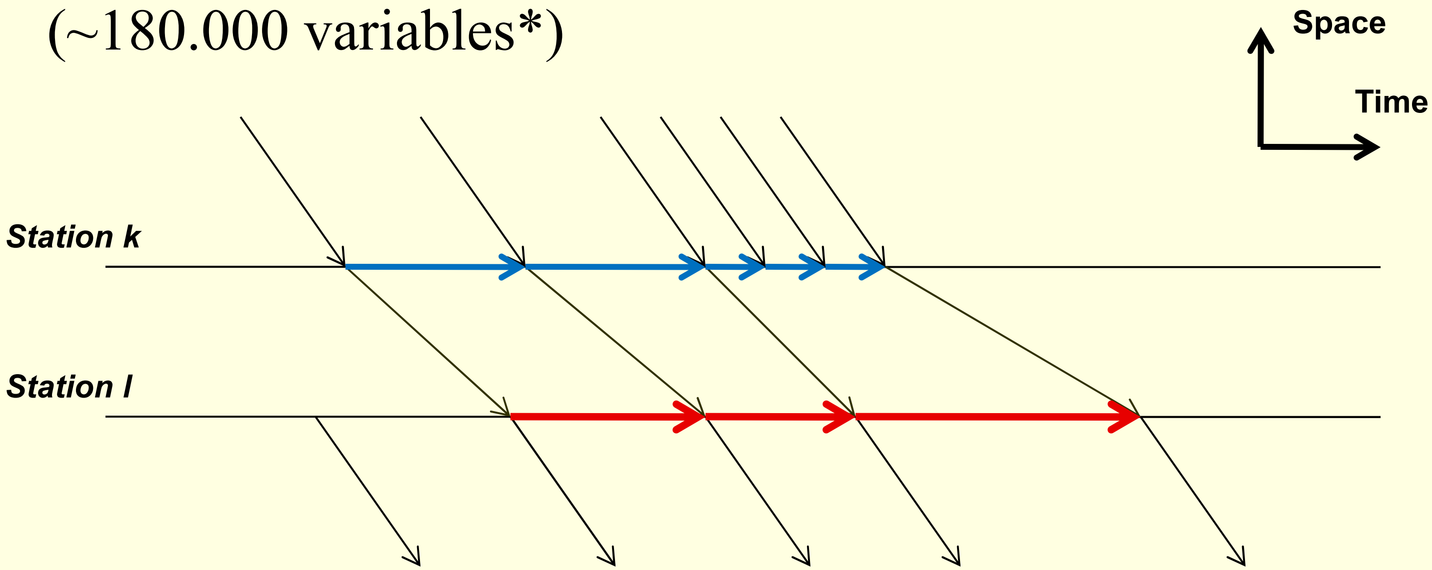
- Introduces timelines
- Nodes are points of time
- Aggregates possible connections into waiting edges
  - Significant decrease in model size (~180.000 variables*)



*Tisza Volán test case

Time space network

- Introduces timelines
- Nodes are points of time
- Aggregates possible connections into waiting edges
  - Significant decrease in model size (~180.000 variables*)



*Tisza Volán test case

MDVSP Solution Methods

- Mathematical methods
  - Exact solution of the IP
  - Column generation
  - Lagrangian relaxation
- Combinatorial
  - Tabu search
- Mixed methods
  - Rounding heuristic
  - Variable fixing (Gintner et al., 2005)

Application oriented (greedy) variable fixing
(Dávid & Krész, 2013)

- Based on the idea of Gintner et al.
- An SDVSP relaxation is solved for the problem
- Consecutive trips of the result are fixed together, if they share some common property:
  - Same number of depots (at least one compatible)
  - Belongs to the same bus line (and have a compatible depot)

Case study:
Szeged urban bus transportation

- Fourth largest city of Hungary
- Population: 168.273
- Bus company: Tisza Volán, Urban Transport Division, www.tiszavolan.hu
- 40 lines, 120 buses
- 4 depots: combinations of
  - Conventional vs articulated (gas fuel)
  - Low floor vs normal

Test results

	Time ratio(%)*	„bad” working pieces**	Cost difference max.(%)
Rounding	50,53	3,5	0,1029
Variable-fixing	14,44	13,75	0,2694
Greedy-chains	7,97	4,75	1,0219

•Comparing to „first feasible exact solution”

**not „driver-friendly”

- Effeciency
  - running time: variable fixing and greedy-chains
  - cost: all heuristics perform well
- Integrity with driver rostering
  - a rounding and greedy-chains

⇒ The greedy-chains heuristics satisfies all aspects

Vehicle Assignment Problem

Scheduling problems in public transportation

Vehicle scheduling

Driver scheduling

Driver rostering

Vehicle assignment
(licence plated buses)

Vehicle assignment problem

- Problem of the VSP:
  - Vehicles in the depots are considered uniform (low floor)
  - Hard to integrate vehicle specific tasks to the model
    - refueling, parking, etc.
- Aim: Solving vehicle scheduling with assigning physical buses
- Satisfy vehicle-specific requirements
  - Refueling
  - Parking
  - Maintenance
- Classical MDVSP models do not support these kind of requirements

Vehicle assignment problem

Solution methods

- Sequential approach
  - Transform an initial vehicle schedule with regards to the vehicle-specific tasks
- Integrated model
  - Build a model for the problem that takes these tasks into consideration
  - It is a 3D asignment model:
  - - lines
  - - „phisical” vehicles
  - - vehicle-specific events

Vehicle assignment problem
Sequential approach
(Árgilán et al, 2013)

Input:

- Set of vehicle schedules
  - Solving an MDVSP model
- Set of vehicles
- Refueling stations with parameters
  - Fuel types, capacity of fuel pumps, opening times
- Fuel pumps:
  - Service times with fixed length
  - Service time may vary depending on fuel type

Vehicle assignment problem
Sequential approach

- Variable:

$$X_{ijkt} = \begin{cases} 1 \text{ if schedule } i \text{ is refueled with vehicle } j \text{ at station } k \text{ in time } t \\ 0 \text{ otherwise} \end{cases}$$

- X_{ijkt} exists (not vorbidden), if
  - The depot *i* corresponds to the depot of vehicle *j*
  - The fuel type *k* corresponds to the fuel type of vehicle *j*
  - Schedule *i* is idle in time period *t*
  - The running distances allow the refueling at time *t*
  - Other conditions can be added

Vehicle assignment problem

Sequential approach

$$\min \sum_{i,j,k,t} c_{ijkt} x_{ijkt}$$

s.t.

$$\sum_{j,k,t} x_{ijkt} = 1 \text{ for all } i, (1 \le i \le n)$$

$$\sum_{i,k,t} x_{ijkt} \le 1 \text{ for all } j, (1 \le i \le m)$$

$$\sum_{i,j} x_{ijkt} \le d_{kt} \text{ for all } k, t(1 \le k \le p), t(1 \le t \le T_d(k))$$

9

Vehicle assignment problem
Sequential approach

- Solve the multi-dimensional assignment problem above
- Problem: „dense” schedules
  - Change for a different bus
  - Remove events
  - New buses

Vehicle assignment problem

Sequential approach – Test cases

Problem	Trips	Depot	Sched. phase	Assign. phase	B	SB	Decr.
Szeged#1	2724	4	1179	14	107	96	8,28%
Szeged#2	2690	4	872	8	107	96	8,28%
Szeged#3	1981	4	431	5	65	54	14,92%
Szeged#4	1768	4	250	1	54	44	16,52%

Vehicle assignment problem
Integrated approach
(Dávid et al, 2014)

Input:

- Set of trips (T)
- Set of different vehicle types (V)
  - common structural parameters:
  - depot, fuel type, capacity, etc.
- Refueling possibilites (R)
  - All the legal time periods, where a vehicle can be refueled
  - capacity k_r for every $r \epsilon R$

Vehicle assignment problem
Integrated approach

- S_v: Set of legal schedules of vehicle v

$$x_s = \begin{cases} 1 \text{ if schedule } s \text{ is carried out} \\ 0 \text{ otherwise} \end{cases}$$

- Variable to connect tasks and schedules

$$a_{is} = \begin{cases} 1 \text{ if task a is carried out in schedule } s \\ 0 \text{ otherwise} \end{cases}$$

- (tasks can be trips, refuelings, or others)

Vehicle assignment problem
Integrated approach

$$\sum_{v=1}^{n} \sum_{s \in S_v} c_s x_s \to min$$

s.t.

$$\sum_{v=1}^{n} \sum_{s \in S_v} a_{ts} x_s = 1 \qquad \forall t \in T$$

$$\sum_{v=1}^{n} \sum_{s \in S_v} a_{rs} x_s \leq k_r \qquad \forall r \in R$$

$$\sum_{s \in S_v} x_s \leq q_v \qquad \forall v \in V$$

$$x_s \in \{0, 1\} \qquad \forall s \in S_v$$

Vehicle assignment problem
Integrated approach

- Solve the problem using column generation
  - Master problem
  - Pricing problem
    - Resource constrained shortest path
    - Check refueling capacities

Vehicle assignment problem

Integrated approach – Test cases

Problem	Trips	Depots	Integrated step	CB	IB	Decr.
Szeged#1	2724	4	4264	107	95	13,34%
Szeged#2	2690	4	3542	107	96	14,22%
Szeged#3	1981	4	2745	65	54	16,04%
Szeged#4	1768	4	2687	54	44	19,63%

„Driver-friendly” Vehicle Scheduling

„Driver-friendly" vehicle scheduling
(Árgilán et al., 2011)

- Schedules given by vehicle scheduling/assignment
- The problem of „dense" schedules
  - Driver rules have to be considered
- Sequential heuristic
  - Uses the results of the previous phases

„Driver-friendly" vehicle schedules

Classification

Class A:  length ≤ M, needs 1 driver

Class B:  M < length ≤ 1,5*M, needs 2 drivers (1,5 drivers actually)

Class C:  1,5*M < length ≤ 2M, needs 2 drivers

„Driver-friendly” vehicle schedules
Transformation

- Cut-and-join for trips of class B
- Split trips of class C
- Different number of break depending on length
- Insertion of breaks
  - No transformation needed
  - Trips have to be removed
    - Can be moved to another schedule
    - Have to be moved to the „free-list”

„Driver-friendly" vehicle schedules

Iteration steps

- New input for vehicle scheduling
  - „Free-list"
- Classification and Transformation on the new schedules
- Do while the „free-list" is empty

„Driver-friendly” vehicle scheduling