Electric Vehicle Routing

MICHAEL FORBES AND JASMINE CRAIG

AMSI OPTIMISE 2017

The problem

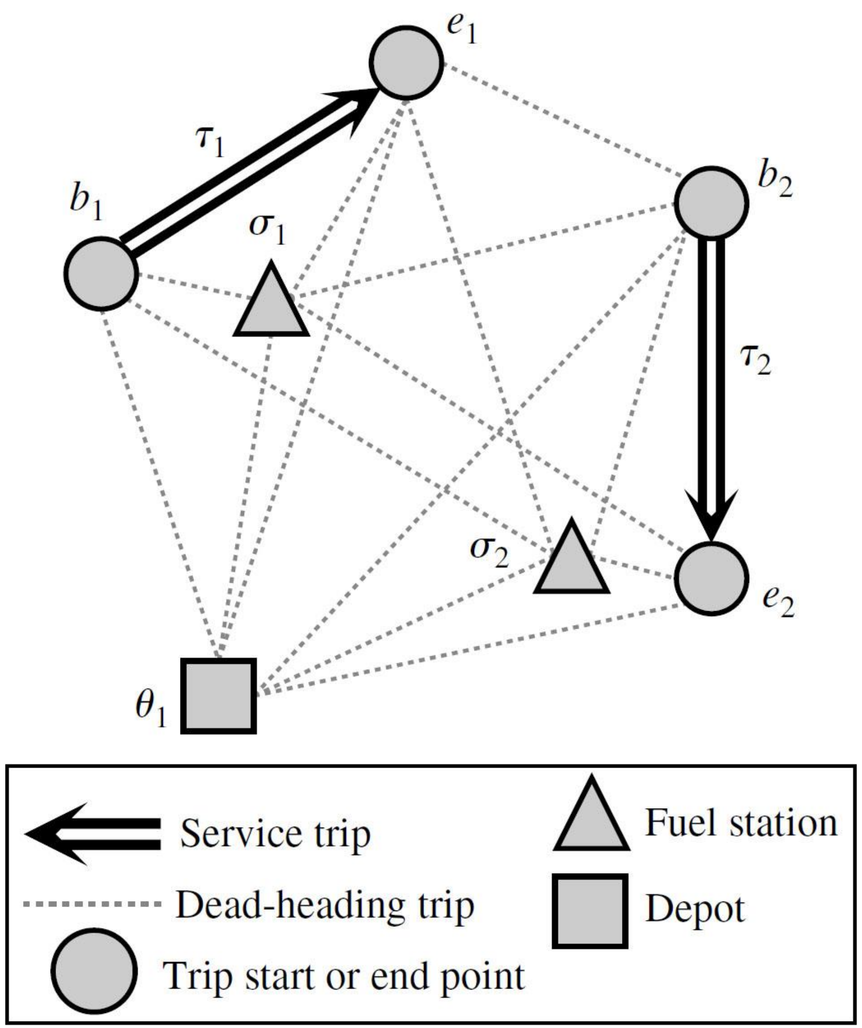
Assign electric buses to a fixed timetable

Minimise cost of owning buses, cost of deadheading and cost of recharging

Subject to:

- Each timetabled trip is operated by exactly one bus
- Each bus returns to its starting depot
- Limit on buses at each depot

- No bus exceeds its range before recharging



The Vehicle Scheduling Problem for Fleets with Alternative-Fuel Vehicles
Adler and Mirchandani, Transportation Science, 2017

Why is this important?

Many cities are switching to electric buses.

But (and perhaps more interestingly)

It’s a recent example of a difficult optimisation problem (an IP) which wasn’t solved well.

The MIP modelling tool kit

Embedded (multicommodity) networks

Composite variables (a priori column generation)

Lazy constraints

Benders Decomposition

Inventory constraints

Discretised time constraints

…

Modelling environment

Discretised time constraints

Tasks operate in at most one discretised time period

Given dependencies between tasks

E.g. mine planning and block dependencies

The "obvious" model:

$$X_{it} = \begin{cases} 1, if\ \ i\ is\ done\ in\ period\ t \\ 0, otherwise \end{cases}$$

$$\sum_t X_{it} \leq 1 \quad \forall i$$

$$X_{it} \leq \sum_{t' \leq t} X_{jt'}\ \forall\ (i, j)\ dependencies$$

Better time constraints

$$Y_{it} = \begin{cases} 1, if\ \ i\ is\ done\ \boldsymbol{by}\ period\ t \\ 0, otherwise \end{cases}$$

$$Y_{it} \leq Y_{i(t+1)} \quad \forall i, t < TMax$$

$$Y_{it} \leq Y_{ij}\ \forall\ (i, j)\ dependencies$$

Bus Scheduling

Easy to optimise if no range limit

- Binary multi-commodity network flow
- “An exact algorithm for multiple depot bus scheduling”
  Forbes, Holt, Watts. European Journal of Operational Research, 1994.

With range limit:

- Primary objective is minimising number of buses
- Secondary is minimising dead heading and recharging time
- Adler and Mirchandani propose Branch and Price and a custom heuristic

Can we do better?

Number of service trips	Number of stations	Number of depots	Mean runtime (sec)	Mean cost (10^4)	Mean number of vehicles	Mean relaxation gap (10^{-3})	Mean number of nodes	Max number of nodes	Mean max level	Max level	Mean number of columns
10	2	2	4	3.52	3.2	0.23	5	13	2.5	5	50
10	4	2	2	3.70	3.4	0.01	2	3	1.5	2	33
10	4	4	2	3.86	3.6	0.03	3	7	1.7	3	53
10	8	4	3	3.65	3.4	0.17	3	5	1.8	3	55
20	2	2	76	6.02	5.4	0.50	147	977	7.2	14	693
20	4	2	17	6.32	5.8	0.38	20	57	4.9	11	174
20	4	4	13	5.84	5.4	0.16	11	43	3.5	6	186
20	8	4	10	5.55	5.2	0.06	6	15	2.8	5	154
30	2	2	89	8.00	7.3	0.28	102	371	8.0	12	736
30	4	2	49	9.15	8.4	0.15	54	271	5.6	13	411
30	4	4	42	8.24	7.7	0.09	26	81	5.1	10	398
30	8	4	34	8.17	7.6	0.07	18	75	4.4	11	307
40	2	2	1,091	10.52	9.5	0.29	1,100	7,001	12.7	20	7,677
40	4	2	183	11.08	10.1	0.21	127	315	8.6	14	974
40	4	4	68	10.01	9.2	0.12	25	91	4.8	11	483
40	8	4	52	9.87	9.1	0.07	12	61	3.0	8	368
50	2	2	3,769	13.43	12.0	0.07	3,577	11,925	16.3	25	16,516
50	4	2	1,616	13.44	12.1	0.05	730	5,835	9.0	22	5,523
50	4	4	1,159	13.49	12.4	0.28	583	4,931	13.0	37	6,188
50	8	4	273	12.77	11.8	0.23	61	141	8.4	15	1,000

A priori generation of fragments

Variables represent:
- a collection of trips that can be done without recharging
- moving between specific recharge locations
- with a bus originating from a specific depot.

Starts at a recharge location (depot) and ends at a recharge location (depot)

Constraints:
- Each trip covered exactly once
- Inventory constraints (by depot) at each recharge location

Pure network flow problems linked by trip coverage constraints

One inventory constraint for each minute (with waiting arcs)

In practice can restrict these to every transition from a departure to an arrival

$$N_t = N_{t^-} + Arrivals\ - Departures$$

Sample results D=2, F=4

Method	N	T_{gen}	T_{build}	T_{solve}	Vars	B&B Nodes	Obj
Exact	50	0.5	1.9	1.4	141794	0	132590
Approx	50	0.3	0.3	0.2	20732	0	133523
Exact	100	3.9	9.7	18.2	704565	0	264490
Approx	100	0.9	1.4	3.5	90725	0	266550
Exact	200	17.6	20.6	39.9	1693819	45	471510
Approx	200	4.5	4.0	4.6	283856	0	477510
Approx	300	51.5	41.3	785.6	2623133	12387	
Approx	400	220.3	151.4	2228.2	9124831	11172	

Can we go even bigger?

Nodes:

- Depot start and depot end
- Recharge locations at epochs
- Ends of trips, replicated for (discretised) charge used by end of trip (e.g. every 10 minutes of range)

Arcs:

- Only connect trips up to a maximum gap (as for approx. algorithm above)
- Connect optimistically (i.e. round down the charge used)
- Replicate arcs for different origin depots

Use lazy constraints to eliminate fragments where rounding down has let through an illegality.

The real problem

There are a lot of papers (and PhDs) published using heuristics where a good MIP model solves to (near) optimality:

- Operating room scheduling, 2016
- The budget-constrained dynamic uncapacitated facility location–network design problem, 2013
- Crane sequencing with yard congestion, 2014
- Twin yard cranes, 2014
- Resource allocation problem in hospitals, 2012
- Truck scheduling in the postal industry, 2017

How do we avoid this?

Yesterday's Email

Transportation Science Articles in Advance

- Column Generation for Outbound Baggage Handling at Airports
- Branch-and-Price-and-Cut for the Active-Passive Vehicle-Routing Problem
  "Computational experiments show that the proposed algorithm delivers improved bounds and solutions for a number of APVRP benchmark instances. It is able to solve instances with up to 76 tasks, four active, and eight passive vehicles to optimality within two hours of CPU time."