Aalto University
School of Science

An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application

Ville Tuominen
Presentation 25
4.12.2020

MS-E2191 Graduate Seminar on Operations Research
Fall 2020

Reference

Simao, H. P., Day, J., George, A. P., Gifford, T., Nienow, J., & Powell, W. B. (2009). *An approximate dynamic programming algorithm for large-scale fleet management: A case application.* Transportation Science, 43(2), 178-197.

Setting

Largest truckload motor carrier in the US that carries customer’s freight

6 000 drivers

Driving tasks may take days

Over 100% turnover: length of employment under a year

Figure 6 **The Schneider Dispatch Center in Green Bay, Wisconsin**

Setting

Drives are dispatched so that empty miles are minimized and the most valuable routes are taken, follow law regarding the drivers and balance profits now and in the future (and possibly other rules)

Drivers have three types of capacity

1. Teams: two drivers in the same tractor who can change
2. Solos. Single drivers who need to rest according to law
3. Independent Contractors: drivers who have their own tractor

Overall 300 different driver types!

Goal

Simulate the long-term truckload operations

- **Closely match the actual behavior the actual operations**
- **Provide marginal value of different types of drivers**

Needs to provide optimal decisions (match drivers to loads) *over time*

- Change in driver allocation has impacts later on

Describe dispatchers by a model

Goal

Possible questions the simulation could answer

- Which people to recruit
- What kind of changes to make to driver pool
- How to price customer contracts
- What freight to take
- How to get drivers home at more predicable time

Method

Deterministic problem not feasible and too limiting

- Large problem
- Uncertainty and unknown relationships

Simulating decisions at single time instants doesn't take future consequences into account

- Example: it would seem optimal to take a long (high-revenue) route to remote city, but taking shorter route to large city with more available loads would be wiser in practice

Approximate Dynamic Programming algorithm

Next

- Notation
- Idea of the algorithm
- Value function approximation
- Marginal values of drivers
- Model calibration

Notation

Driver attributes	$\boldsymbol{a} = (\boldsymbol{a_1}, \boldsymbol{a_2}, \dots) =$ (location, home city, capacity, ...)	
Load attributes	$\boldsymbol{b} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \dots) =$ (destination, revenue, priority, ...)	
Number of resources	$\boldsymbol{R_t} = (\boldsymbol{R_{ta}})$	*state of drivers*
Number of loads	$\boldsymbol{D_t} = (\boldsymbol{D_{tb}})$	*state of freight*
Change in numbers	$\widehat{\boldsymbol{R}}_t, \widehat{\boldsymbol{D}}_t$	
System state	$\boldsymbol{S_t} = (\boldsymbol{R_t}, \boldsymbol{D_t})$	
New information	$\boldsymbol{W_t} = (\widehat{\boldsymbol{R}}_t, \widehat{\boldsymbol{D}}_t)$	
Value function approximation	$\overline{\boldsymbol{V}}_t^n$	
Contribution	$\boldsymbol{C_t} =$ **profits − costs − penalties for late delivery + bonuses + …**	

Algorithm: Idea

Iteratively simulate system forward in time

- Calculate value function for each driver at each *possible* state → value of a driver at end of each task
- Use that to decide how to allocate drivers to loads

The optimal policy maximizes the expected sum of contributions

Algorithm: Idea

Solve assignment of drivers to loads: x_t^n

$$\max_{x_t} C_t(S_t^n, x_t) + \gamma \overline{V_t^{n-1}}(S_t^X)$$

Contribution Discount Value function approximation from previous iteration

and update value function approximation $\overline{v}_{t-1}^n$

Aalto University
School of Science

Step 0: Initialization:

Step 0a: Initialize $\overline{V}_t^0,\ t \in \mathcal{T}$.

Step 0b: Initialize the state S_0^1.

Set initial state

Step 0c: Set $n = 1$.

Step 1: Choose a sample path ω^n.

Sample path determines R and D

Loop over n

Loop over t

Step 2: Do for $t = 0, 1, \ldots, T$:

Step 2a: Solve the optimization problem:

$$\max_{x_t \in \mathcal{X}_t^n} \left(C_t(S_t^n, x_t) + \gamma \overline{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right).$$

Let x_t^n be the value of x_t that solves Equation (20), and let ν_{ta_t} be the dualcorresponding to the resource conservation constraint for each R_{ta_t} where $R_{ta_t} > 0$.

Assignment of drivers

Step 2b: Update the value function using

$$\bar{v}_{t-1,a}^n = (1 - \alpha_{n-1})\bar{v}_{t-1,a}^{n-1} + \alpha_{n-1}\hat{\nu}_{ta}^n.$$

Do this for each attribute a for which we have computed $\hat{\nu}_{ta}^n$.

Update value function for all a

Step 2c: Update the state:

$$S_t^{x,n} = S^{M,x}(S_t^n, x_t^n)$$
$$S_t^n = S^{M,W}(S_{t-1}^{x,n}, W_t(\omega^n)).$$

Update post-decision states

Step 3: Increment n. If $n \le N$, then set $S_0^{x,n} = S_T^{x,n-1}$ and go to Step 1.

Set new initial state for next iteration

Step 4: Return the value functions, $\{\bar{v}_{ta}^n, t = 1, \ldots, T, a \in \mathcal{A}\}$.

Algorithm: Value Function Approximation

To evaluate driver marginal values, we use the value function

Calculating the value function for each attribute is too large problem

Luckily not all attributes are needed for approximating value function.

- We need only driver location, driver home city and driver capacity type
- Results in 600 000 attributes

In addition, we can *aggregate* attributes to smaller resolution

- For example from 100 regions to 10 regions
- Results in 6 000, 2 000 and 200 attributes with 3 different aggregations

Algorithm: Value Function Approximation

The estimate of value function is the weighted sum of the values at different aggregation levels

$$\overline{v_{ta}} = \sum_{g} w_{ta}^{(g)} \overline{v}_{ta}^{(g)}$$

Weight depends on

- **aggregation level g**
- **current index of iteration n**

$$w_{ta}^{(g,n)} = \left(\underbrace{\left(\sigma_{ta}^2\right)^{(g,n)}}_{\text{Estimate of variance}} + \underbrace{\left(\mu_{ta}^{(g,n)}\right)^2}_{\text{Estimate of bias due to aggregation}} \right)^{-1}$$

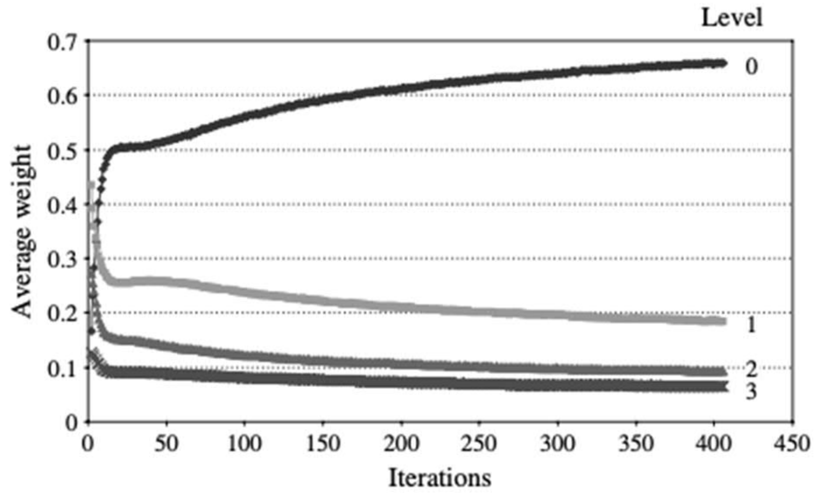


Figure 3 **Average Weight Put on Each Level of Aggregation by Iteration**

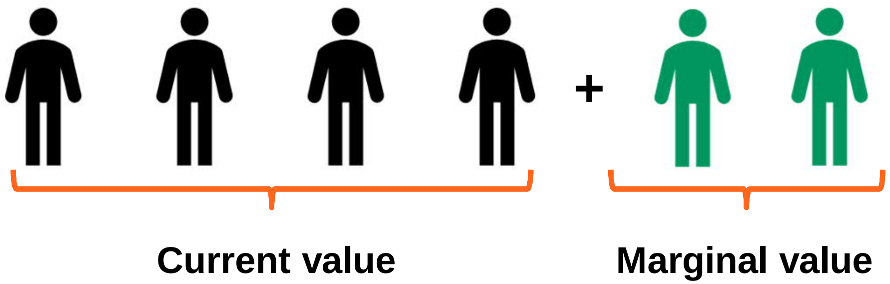
Marginal values of driver types

Value of adding driver at certain location (or other actions)

Mimic dual variables of Linear Program

Value function approximations provide accurate estimate: the result of simulation of time $t = 0$: $\overline{v_0}^N$

No need to run any simulation again when examining the impacts of changes



Calibration

The truck company wanted that the model closely replicates multiple variables historic data, since for example driver turnover was not modelled otherwise

Most importantly match patterns on

- Average length of haul for different driver capacity types
- Percentage of drivers not at home on weekend

to the decisions from dispatchers

Company gave the intervals where values should fit

Calibration: Pattern Matching

Pattern: represents percentage of time that *resources with attribute a* are acted on by *decisions d*
We aggregate attributes and decisions into categories

Objective function with patterns

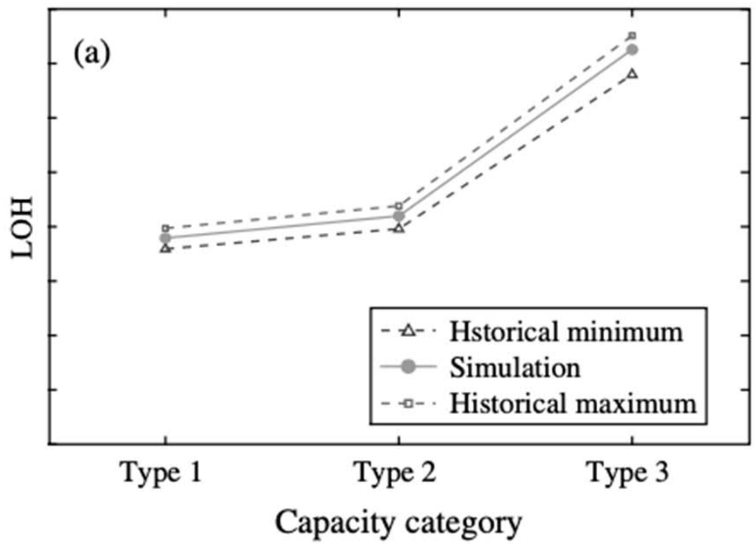
Pattern weight

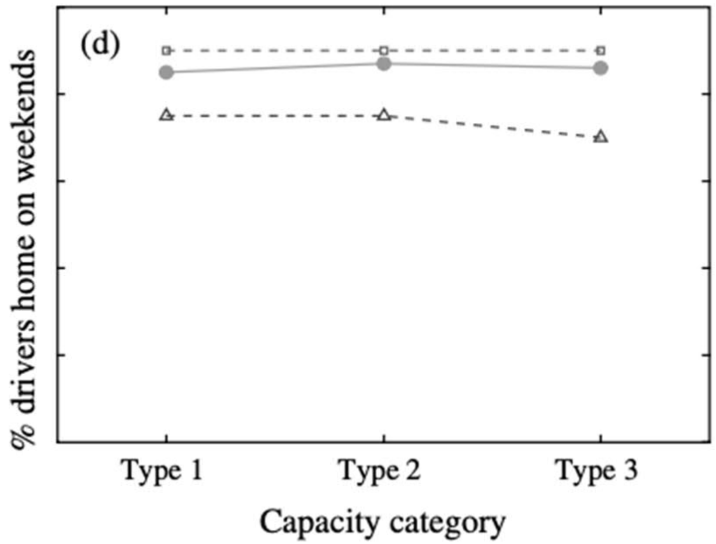
$$x_t(\theta) = \arg\max_{x_t} \left[C_t(S_t^n, x_t) - \theta H(\rho(x), \rho^{\varrho}) \right]$$

Contribution

Deviations from observed frequencies (square error)

Calibration

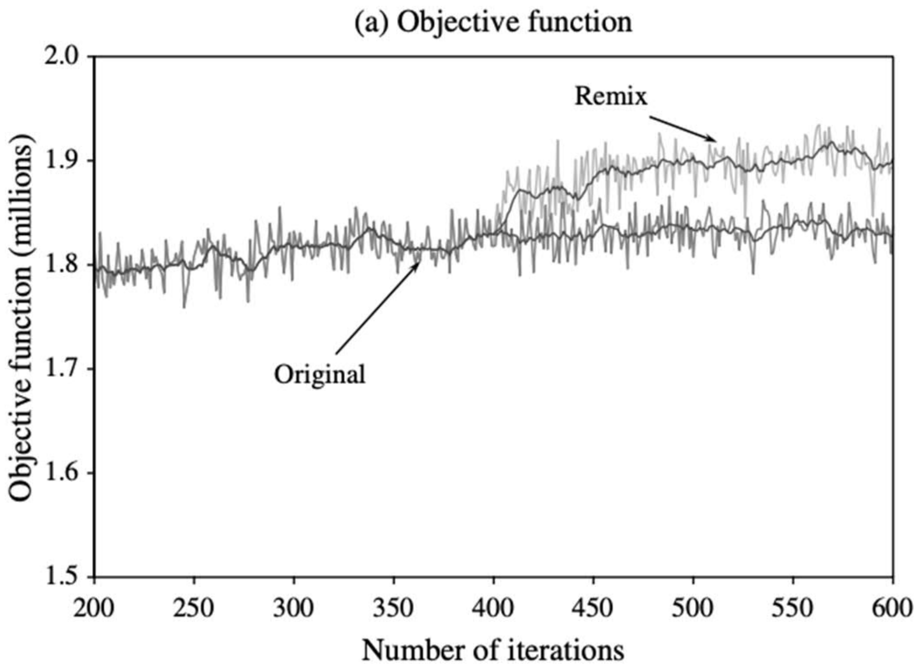




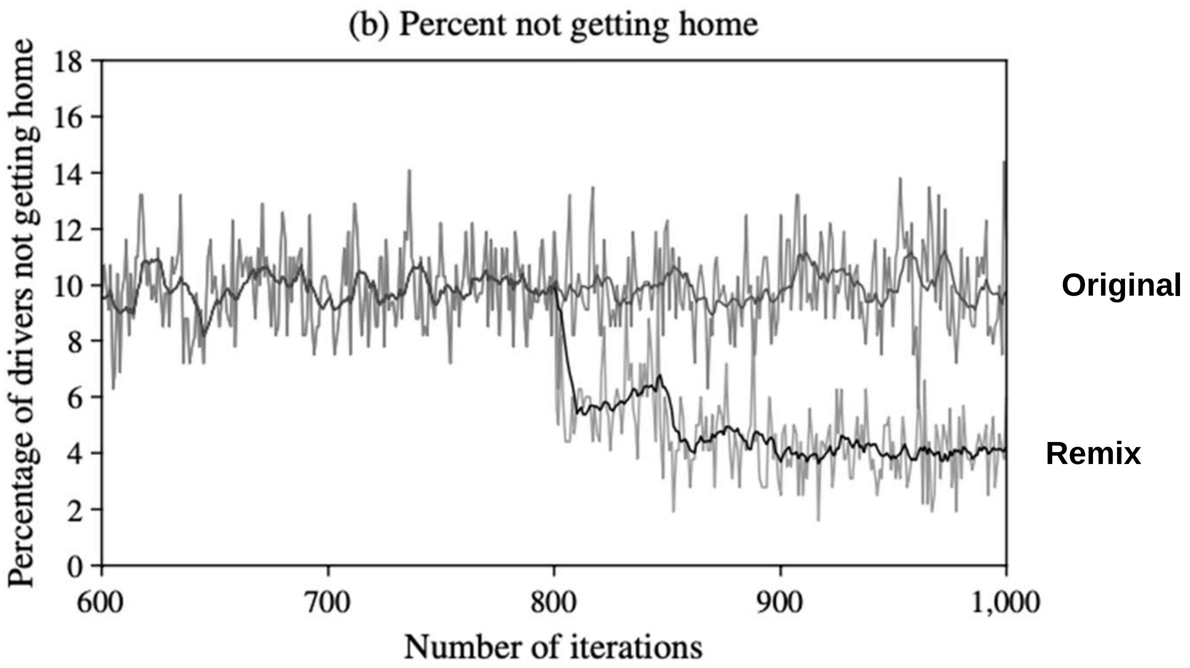
Pattern matching used for length of haul, other variables using cost-based rules

Results: Fleet Mix

Redistribute the drivers between the three classes (team, solo, contractor)



Results: Fleet Mix



Aalto University
School of Science

Results: Benefits

- **93 % self-scheduling flexibility for drivers that cost *only $6 million* instead of initial plan of $30 million.**
- **Quantifying the cost of hours-of-service rules results more power to negotiate with customers. *Margin improvements of 2-3%.***
- **Better scheduling*. Margin improvement of 4-10%* and 50% less late deliveries.**
- **Reduce number of drivers crossing borders. *$4 million savings* in training etc.**
- **Better understanding of effects of hiring. *$5 million profit* annually**

Homework

Aalto University
School of Science

Homework

Read section *5. Fleet Mix Studies* from the article and briefly report

- How well the model estimates marginal values of drivers (5.1)
- How the Driver Remix Experiment was done (5.2.)

DL 11.2., submit to ville.m.tuominen[at]aalto.fi

Forward vs backward

Fowrad: starts from 0 and slowly rises: value may be underestimate. Value depends on previous *iteration*

Two-pass: forward without updating the value functions. Derivative of value function is the computden in a backward pass. Value depends on following *time*

In this problem we can't use standard DP for the backward pass. Instead calculate numric derivative for each driver by removing one driver at the time. This gives the value at backard pass

Using dual variables instead of numerical derivates (much faster and approximately same dispatch decisions, altough less accurate value functions

Outline

Algoritmi

Dual

Kalibrointi

Kotitehtävä

- Askelkoko?
- Pattern matching ja VFA vaikutus mallin suoriutumiseen?
- Driver remix?

Homework

Kotitehtävä

- Askelkoko?
- Pattern matching ja VFA vaikutus mallin suoriutumiseen?
- Driver remix?

Some thoughts

Development took three years

- Made with research team at Princeton University
- Resulted 3 PhD dissertations
- Intense support from the truck company

→ Real risk in development

- *(let's imagine 3x data scientists + 3x students for three years + resources from company)*

In this case the result was high profits

- Do the University or the scientists get paid enough?
- While the driver satisfaction was also considered (and increased) is this still (morally) optimal solution (as the turn rate is so high)?