Choice set generation for iterated DTA simulations

Gunnar Flötteröd and Michel Bierlaire

May 4, 2011

Introduction

- a paths is a connected sequence of nodes in a network
- concept of “path” carries over to “travel plan”
- DTA simulations: huge (path) choice sets
- objective: efficient path sampling from arbitrary distributions

Outline

Outline

Relevance of choice set modeling

The Metropolis-Hastings algorithm

Metropolis-Hastings sampling of paths

Simple example

Tel-Aviv example

Outlook & summary

Choice process

- choice process: decision maker n...
    1. considers a set $\mathcal{C}_n$ of alternatives
    2. selects one alternative i from that set
- two modeling questions:
    1. what choice set $\mathcal{C}_n$ is considered?
    2. given $\mathcal{C}_n$, what choice i is made?

Choice set and choice

- choice in the presence of an uncertain choice set

$$P_n(i) = \sum_{\mathcal{C}_n \subseteq \mathcal{C}} P_n(i|\mathcal{C}_n) P_n(\mathcal{C}_n)$$

- simulation: draw from $P_n(i)$ by
  1. drawing $\mathcal{C}_n$ from $P_n(\mathcal{C}_n)$
  2. drawing i from $P_n(i|\mathcal{C}_n)$
- the choice set is decisive for the simulated choice!

Modeling of path choice sets

- difficult because real choice set is typically not observable
- two broad classes of methods
  - modeling of consideration sets
    - deterministic (e.g., K-SP) or stochastic (randomized SP)
    - unrealistic: fail to capture the chosen alternative
  - assume that decision maker considers all alternatives
    - also unrealistic
    - sampling protocol generates operational subset
    - correct for sampling in the estimation

Outline

How to sample from large (path) choice sets?

- approach
  - give every path $i \in \mathcal{C}$ a weight $b(i) > 0$
  - sampling probability $q(i)$ shall be $\propto b(i)$
- direct sampling from $q(i)$ requires path enumeration

$$q(i) = \frac{b(i)}{\sum_{j \in \mathcal{C}} b(j)}$$

- but pair-wise comparison of paths is easily done

$$\frac{q(i)}{q(j)} = \frac{b(i)}{b(j)}$$

Metropolis-Hastings (MH) algorithm

1. set iteration counter $k = 0$
2. select arbitrary initial state i^k
3. repeat beyond stationarity
    3.1 draw candidate state j from **proposal distribution** $q(i^k, j)$
    3.2 compute **acceptance probability**

$$\alpha(i^k, j) = \min\left(\frac{b(j)q(j, i^k)}{b(i^k)q(i^k, j)}, 1\right)$$

    3.3 with probability $\alpha(i^k, j)$, let $i^{k+1} = j$; else, let $i^{k+1} = i^k$
    3.4 increase k by one

Convergence of MH algorithm

- given
  - a finite state space
  - positive weights $b(i)$
  - an irreducible[1] proposal distribution $q(i,j)$

  MH converges to stationary distribution[2] $b(i)/\sum_j b(j)$
- proposal distribution $q(i,j)$ crucial for convergence speed
  - too little variability: slow convergence
  - too much variability: random search

[1] every state can (eventually) reach every other state
[2] long-term state coverage of the process

Outline

State space

- a state $i = (\Gamma, a, b, c)$ consists of
  - a path Γ
  - three node indices $a < b < c$ within that path
- node indices simplify computation of transition probabilities

Weights

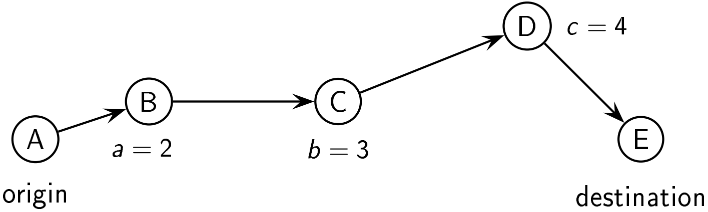
- intuitive: weight $\exp[-\mu\delta(\Gamma)]$ with path cost $\delta(\Gamma)$ and $\mu \geq 0$
- there are $|\Gamma|(|\Gamma|-1)(|\Gamma|-2)/6$ states with the same Γ
- corrected weights:

$$b(i) = \frac{\exp[-\mu\delta(\Gamma)]}{|\Gamma|(|\Gamma|-1)(|\Gamma|-2)/6}$$

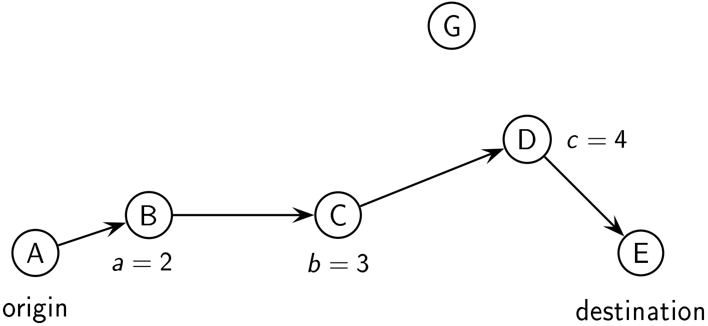
Proposal distribution

- SHUFFLE operation
  - re-sample (uniformly) $a < b < c$ within path Γ
- SPLICE operation
  - sample a node v "near" the path segment $\Gamma(a) \ldots \Gamma(c)$
  - connect $\Gamma(a)$ to v
  - connect v to $\Gamma(c)$
  - let new b point at v, update c
- combined proposal: randomly select one procedure
- [[complicated computation of proposal probabilities]]

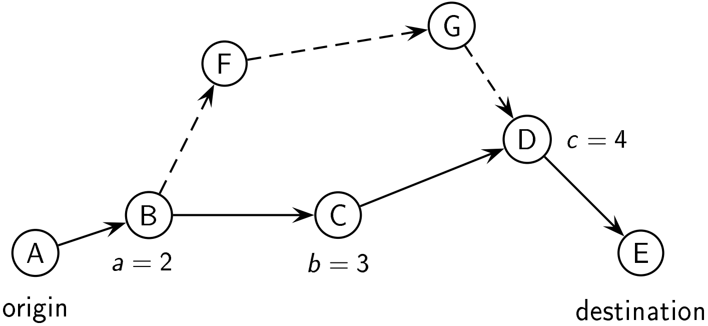
SPLICE example



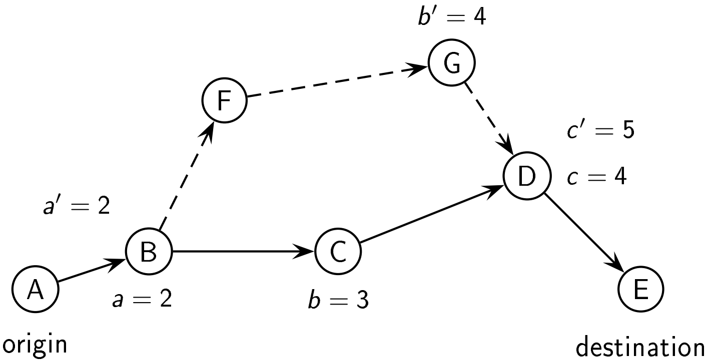
SPLICE example



SPLICE example

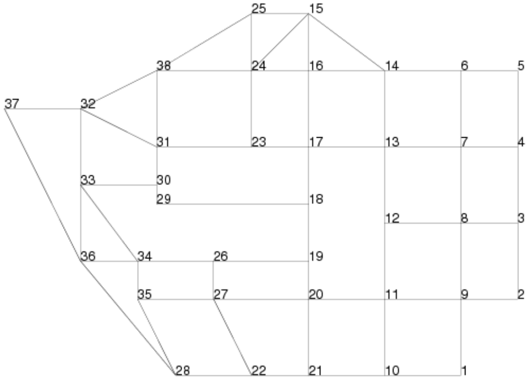


SPLICE example

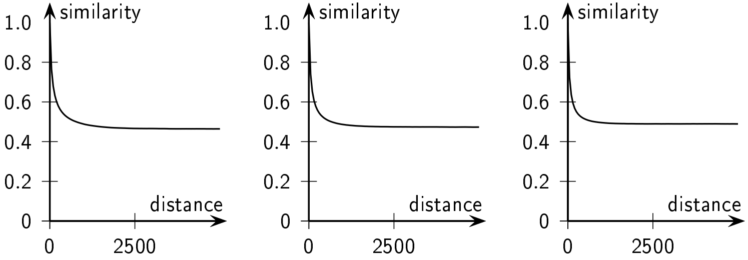


Outline

Simple example



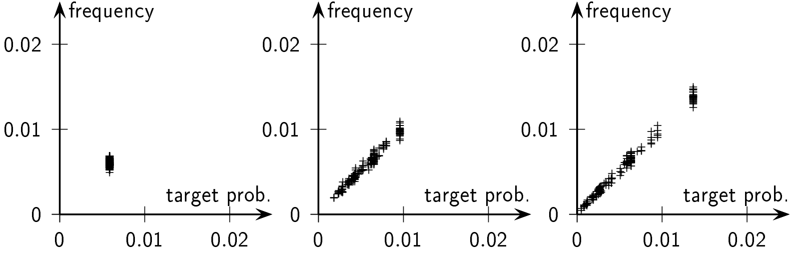
Simple example: correlation within the chain



(a) $\mu = 0.0$ (b) $\mu = 2.0$ (c) $\mu = 4.0$

- for independent draws, extract every 2500th path

Simple example: scatterplots



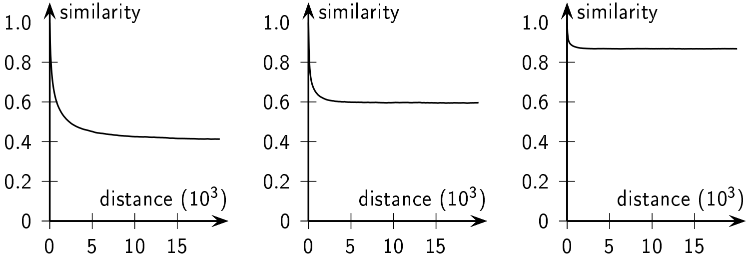
(a) $\mu = 0.0$ (b) $\mu = 2.0$ (c) $\mu = 4.0$

- χ^2 test does not reject hypothesis: sample from target distr.
  - test statistics: 198.93, 177.29, 157.69 for $\mu = 0, 2, 4$
  - 0.5, 0.9, 0.95 quantiles: 168.33, 192.95, and 200.33

Outline

Tel-Aviv example: network

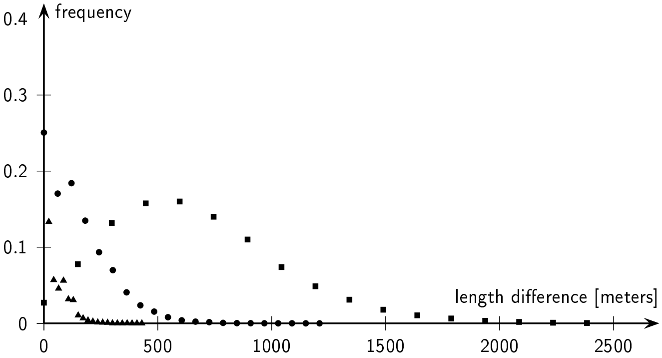
Tel-Aviv example: within-chain correlation



(a) $\mu = 0.01$ (b) $\mu = 0.02$ (c) $\mu = 0.04$

- for independent draws, extract every 10 000th route

Tel-Aviv example: length distribution



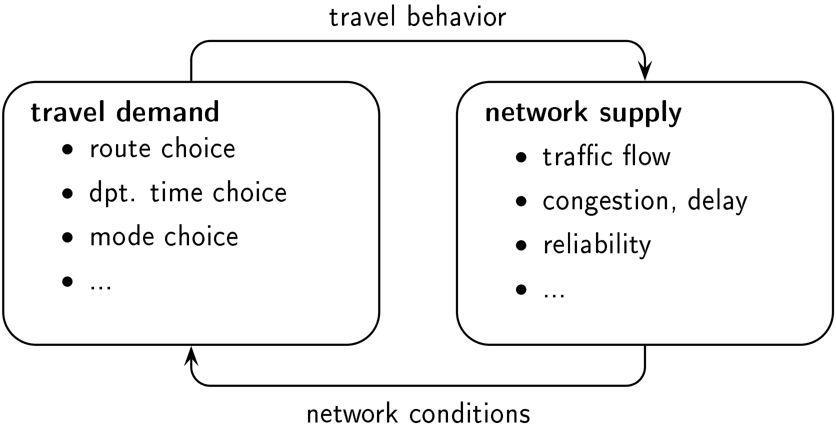
Squares: $\mu = 0.01$, circles: $\mu = 0.02$, triangles: $\mu = 0.04$.

Computational performance

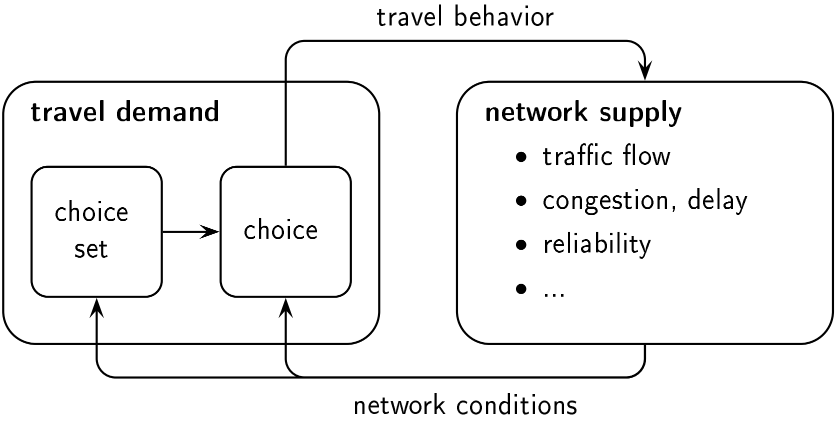
- performance is quite problem specific
  - narrow target distribution → faster
  - origin/destination nearby → faster
  - small overall network (or preprocessing) → faster
  - missing some routes uncritical→ faster
  - simple proposal distribution → probably faster
- Tel-Aviv example: order of $10^3 \ldots 10^4$ iterations per minute
- main bottleneck: *many* shortest path tree computations

Outline

Outlook: choice set formation in the simulation loop



Outlook: choice set formation in the simulation loop



Summary

- Metropolis-Hastings sampling of paths
  - generalizes to all-day travel plans
  - well-specified choice set distributions
- operational implementation
  - upcoming re-implementation of BIOROUTE
  - quite efficient (but can be tuned further)
- consistency with iterated DTA simulations
  - iterated simulation constitutes Markov chain
  - so does Metropolis-Hastings algorithm
  - specification of one joint chain