Stochastic optimization of simulation models by inertial stochastic finite differences

Alexei A. Gaivoronski

Department of Industrial Economics and Technology Management
Norwegian University of Science and Technology
Alexei.Gaivoronski@iot.ntnu.no

Parts of this lecture present a joint work with
Denis Becker, Trondheim Business School, Norway
Paola Zuddas, University of Cagliari, Italy

Stochastic optimization of simulation models

- Important theme of Professor Pflug research
- Shows his vision how to select a difficult theme that is destined to have a lasting and increasing significance for theory and practice of optimization
- He has obtained some of the most penetrating and fundamental results that continue to shape the field
- He has produced one of the most authoritative texts:

Georg Ch. Pflug. *Optimization of Stochastic Models. The Interface Between Simulation and Optimization*, Kluwer, Boston, 1996

Stochastic optimization of simulation models

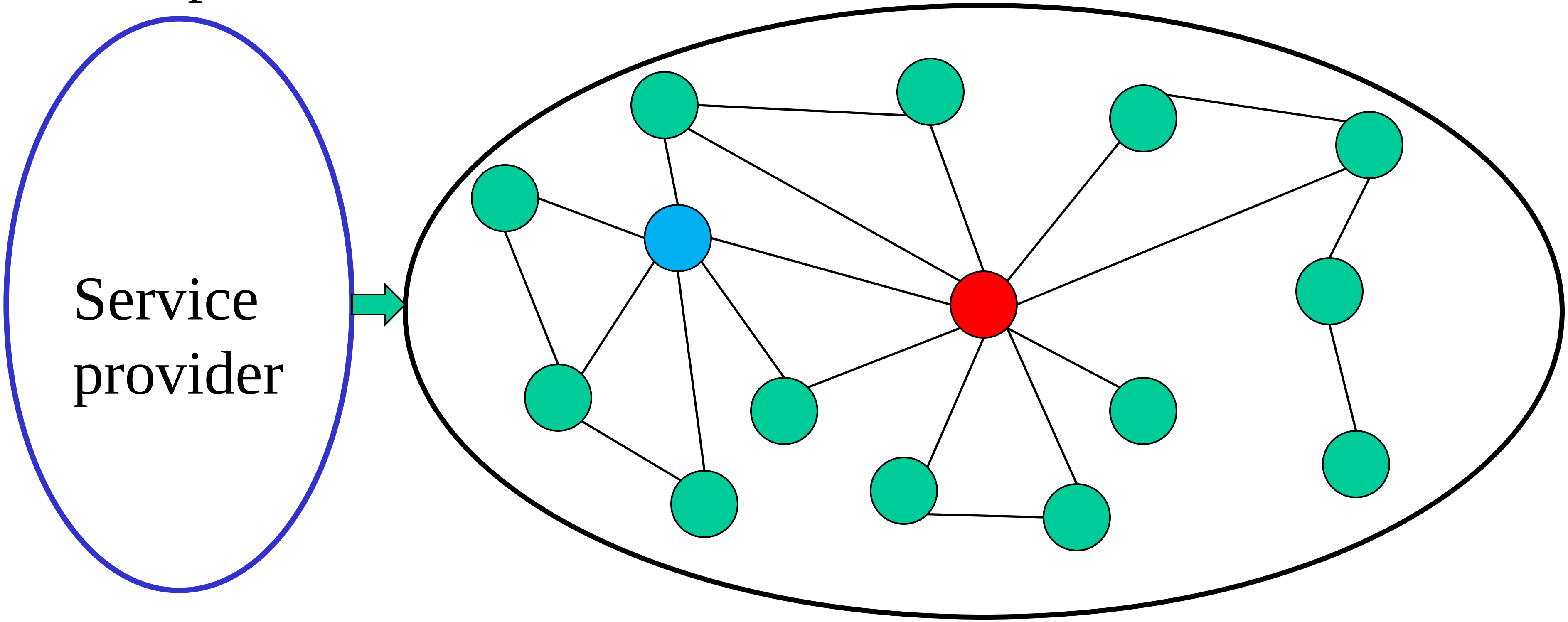
- Why it will be more and more useful:
  - Many real systems that operate in conditions of uncertainty can not be described by a relatively limited collection of linear (and even nonlinear) equations without making serious tradeoffs between computability and adequacy
  - Simulation is a natural paradigm in such situations
  - Decision policies are naturally modelled as functions of relatively limited number of parameters of state of the model (hundreds and not billions)
  - BUT, computational requirements are serious, brute force does not work, intelligent algorithms are needed. With right algoritms computational power is already here, 10^7 iterations are feasible.
  - Examples: ICT, production, transportation, supply chain

Contents

- Two examples:
  - Differentiated service pricing on social network
  - Transportation between supply/demand nodes with inventories
  - **Common feature**: complex networks where small cases are solvable with normative approach, but optimization of simulation model allow to arrive much, much further
- Numerical method: stochastic inertial finite differences
  - Avoids different pitfalls that often happen along this road
- Some insights from numerical experiments

Example 1: Differentiated service pricing on social network

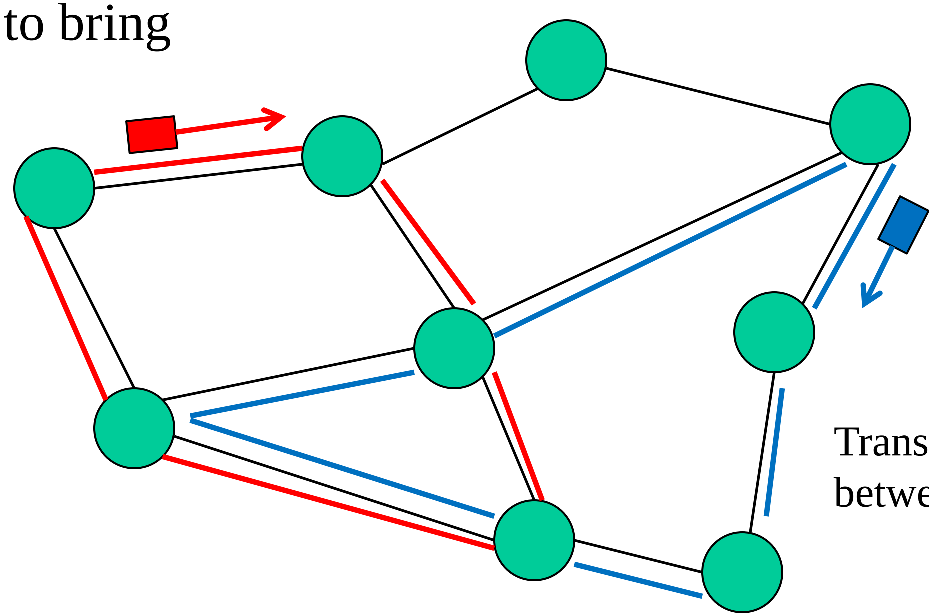
- Service provider who maximizes his profit
- Population of customers connected in social network



Social network: nodes with links that randomly change in time

Example 2: Transportation between supply/demand nodes with inventories

- Set of nodes connected with links (roads, waterways …) with supply/demand (or both) for a collection of items
- Transportation (trucks, ships) operates on links that takes items in excess and bring them where they are lacking
- Transportation time, finite inventories, costs of transportation, inventory, backlog
- Decisions taken before one step before inventory is known: where to take and where to bring



Transportation of empty containers between ports

NTNU
Alexei.Gaivoronski@iot.ntnu.no

Differentiated service pricing on social network

- How to price a service offered to participants in a social network in order to maximize profit?
- Uniform price? Or maybe differentiated price? Offer discounts to well connected participants? If yes, how much?

We develop model and integrated simulation and optimization tool that answers these questions

Model of social network

- Set of nodes $i=1:N$
- Discrete time $t=1,\ldots,T,\ldots$
- $A^t(w)$ – incidence process: a stationary Markov process with values in the space of NxN matrices; the element a^t_{ij} with value between 0 and 1 describes the ”strength” of connection between nodes i and j.
- $B_i^t(\omega) = \{j | a_{ij}^t(\omega) > 0\}$ the set of neighbors of node i at time t
- *Service provider (SP)* provides service described by set of parameters x (subscription and unit usage prices for different user categories, QoS, SLA, …) decided by SP. SP can monitor usage and structure of the network (possibly with errors).

Model of social network, continued

- *Customers* decide about subscription y_i^t (0 or 1) and volume of usage z_i^t.
- Subscription y_i^t of user i at time $t+1$ depend on
  - Current subscription decision y_i^t
  - Service parameters x
  - Knowledge about the service usage among network neighbors from $B_i^t(\omega)$: *the network effect function* $v_i^t(\omega)$

$$v_i^t(\omega) = v_i\left(\left(y_j^t, z_j^t\right) | j \in B_i^t(\omega)\right)$$

  - Random event ω
  - Thus, subscription process is defined as follows:

$$y_i^{t+1} = y_i^{t+1}\left(y_i^t, x^t, v_i^t(\omega), \omega\right)$$

and more specifically by transition probabilities

$$\mathbb{P}\left\{y_i^{t+1} = k \mid y_i^t = l\right\} = p_{ti}^{kl},\ p_{ti}^{kl} = p_{ti}^{kl}\left(x^t, v_i^t(\omega), \omega\right),\ k, l = 0, 1$$

Alexei.Gaivoronski@iot.ntnu.no

Model of social network, continued

- The volume of traffic z_i^t of customer *i* at time *t* also depends on the service parameters and network effects:

$$z_i^t = \begin{cases} z_i^t\left(x^t, v_i^t(\omega), \omega\right) > 0 & \text{if} \quad y_i^t = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Revenue of SP from user *i* is defined by subscription and traffic: $r_i^t = r_i^t\left(x^t, y_i^t, z_i^t\right)$
  – Example: $x=(x_{i1},x_{i2})$ where x_{i1} is the subscription flat rate and x_{i2} is unit traffic rate for user *i*, then

$$r_i^t = \begin{cases} x_{1i} + x_{2i} z_i^t & \text{if} \quad y_i^t = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Total revenue during the time horizon *1,…,T*:

$$R^T(x,\omega) = \sum_{t=1}^{T} R_t(x,\omega), \;\; R_t(x,\omega) = \sum_{i=1}^{N} r_i^t\left(x, y_i^t, z_i^t\right)$$

Revenue maximization on social network

- Decision problem od service provider:
  - Find service parameters *x* that maximize *F(x)*

$$F(x) = \lim_{T\to\infty} \frac{1}{T} R^T(x,\omega)$$

under constraints $x \in X$ where X is admissible set.

This is very nontrivial optimization problem.

Difficulties: randomness and transient behavior, precise values of objective function can be obtained only after long simulation process

Alexei.Gaivoronski@iot.ntnu.no

Possible optimization strategies: many pitfalls here

Sample path optimization – often advocated but will fail here

1. Select very long time horizon T
2. Generate all the random numbers ω^* necessary for simulation during this time horizon
3. Use off-shelf nonlinear optimization software to solve the problem

$$\max_{x \in X} \frac{1}{T} R^T(x, \omega^*)$$

4. Advantage: no effort is needed for implementation and tuning of optimization algorithm
5. Drawback: will not work due to highly irregular behavior of the sample path objective

Example A

- Fixed constant incidence matix

$$A = \left\| \begin{array}{cccccc} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right\|$$

- Network effect function is between 0 and 1; the more users from the neighborhood of user i are subscribed the closest it is to 1

$$v_i^t(\omega) = 1 - e^{-\eta_i \sum_j a_{ij} y_j^t}$$

- The single service parameter is its price *x* per unit of traffic, the same for all users

Example A, continued

- Subscription/unsubscription probabilities are deterministic functions of price and network effect

$$p_{ti}^{01} = \left(1 - \frac{\min\{x, p_{i\max}\}}{p_{i\max}}\right)\left(b_i + (1 - b_i)\, v_i^t\right), \; p_{ti}^{00} = 1 - p_{ti}^{01},$$

$$p_{ti}^{10} = \frac{\min\{x, p_{i\max}\}}{p_{i\max}} (1 - q_i)\, v_i^t, \; p_{ti}^{11} = 1 - p_{ti}^{10}$$

where $p_{i\max}$ is the maximal price the user *i* is willing to accept and b_i, q_i are some parameters between 0 and 1

- The traffic function

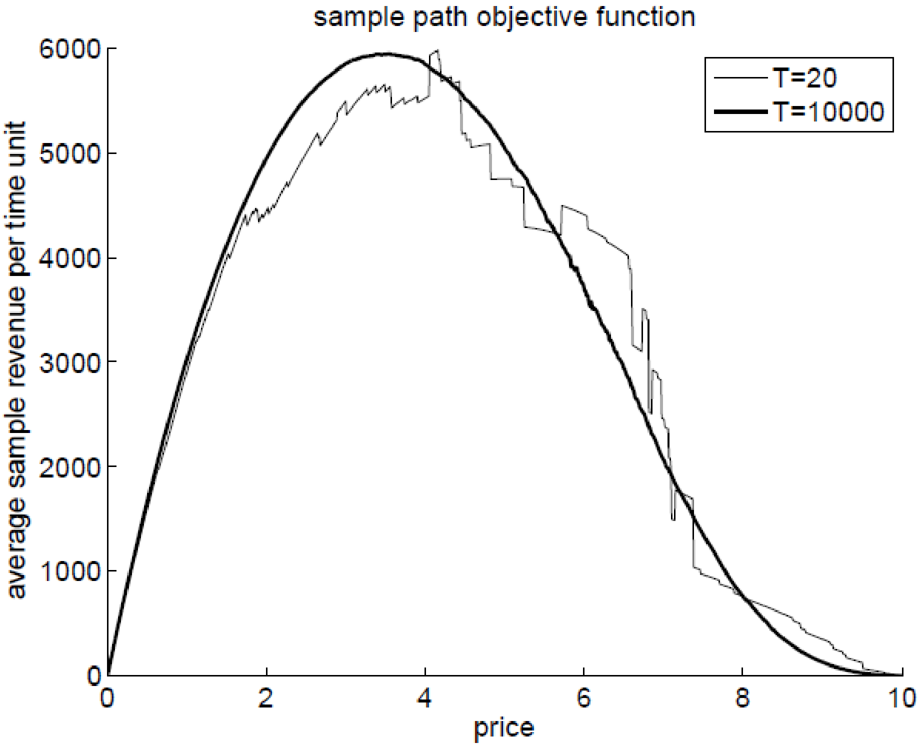
$$z_i^t = \begin{cases} \left(1 - \frac{\min\{x, p_{i\max}\}}{p_{i\max}}\right)\left(c_{1i} + c_{2i} v_i^t\right) & \text{if} \quad y_i^t = 1 \\ 0 & \text{otherwise} \end{cases}$$

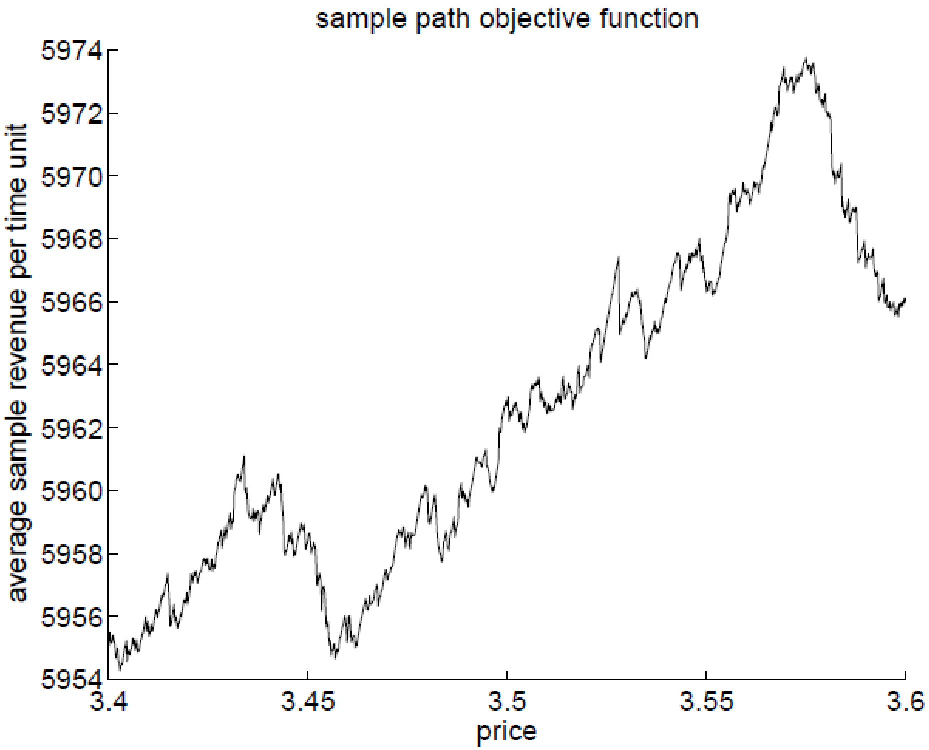
- Revenue $r_i^t = x z_i^t$

Alexei.Gaivoronski@iot.ntnu.no

Example A, continued

- Sample path objective functions for T=20 and T=10000





- Irregular behavior persists irrespective of the length of the time horizon, local maxima are just about everywhere. Off-shelf optimization codes will fail

Answer: stochastic (quasi)gradient methods (SQG)

- Method for solution of stochastic optimization problems of the type

$$\max_{x\in X} \mathbb{E} f(x,\omega)$$

- Iterative process $x^{s+1} = \pi_X\left(x^s + \rho_s \xi_s\right)$
- Projection operator

$$\pi_X(y) \in X, \ \ \|\pi_X(y) - y\| = \min_{z\in X} \|z - y\|$$

- ξ_s is an estimate of the gradient of the objective function:

$$\mathbb{E}\left(\xi_s \mid x^1, \ldots, x^s\right) = F_x\left(x^s\right) + a_s$$

- ρ_s is a stepsize:

$$\sum_{s=0}^{\infty} \rho_s = \infty, \ \sum_{s=0}^{\infty} \rho_s^2 < \infty, \ \sum_{s=0}^{\infty} \rho_s \|a_s\| < \infty \ \ a.s.$$

Stochastic approximation: Kiefer & Wolfowitz
Professor Pflug has made important contributon here too

SQG, continued

- Important difference: transient behavior, we can observe only estimate with the property

$$\mathbb{E}\left(\xi_s \mid x^1, ..., x^s\right) = F_x^s\left(x^s\right) + a_s$$

- **Convergence theorem**:
  - Functions $F^s(x)$ are convex and bounded on open set
  - Set X is convex and compact
  - ρ_s is nonnegative and

$$\sum_{s=0}^{\infty} \rho_s = \infty, \ \sum_{s=0}^{\infty} \rho_s^2 < \infty, \ \sum_{s=0}^{\infty} \rho_s \left\| a_s \right\| < \infty \ \ a.s., \ \mathbb{E}\left(\left\| F_x^s\left(x^s\right) - \xi_s - a_s \right\|^2 \ \mid \mathbb{B}_s \right) < \infty$$

  - Nonstationarity condition:

$$\frac{\sup_{x \in X} \left\| F^{s+1}(x) - F^s(x) \right\|}{\rho_s} \rightarrow 0 \ \ as \ \ s \rightarrow \infty$$

Then $\max_{x \in X} F^s(x) - F^s\left(x^s\right) \rightarrow 0$ with probability 1

Alexei.Gaivoronski@iot.ntnu.no

Lot of work is needed to adapt this method to optimization of simulation models

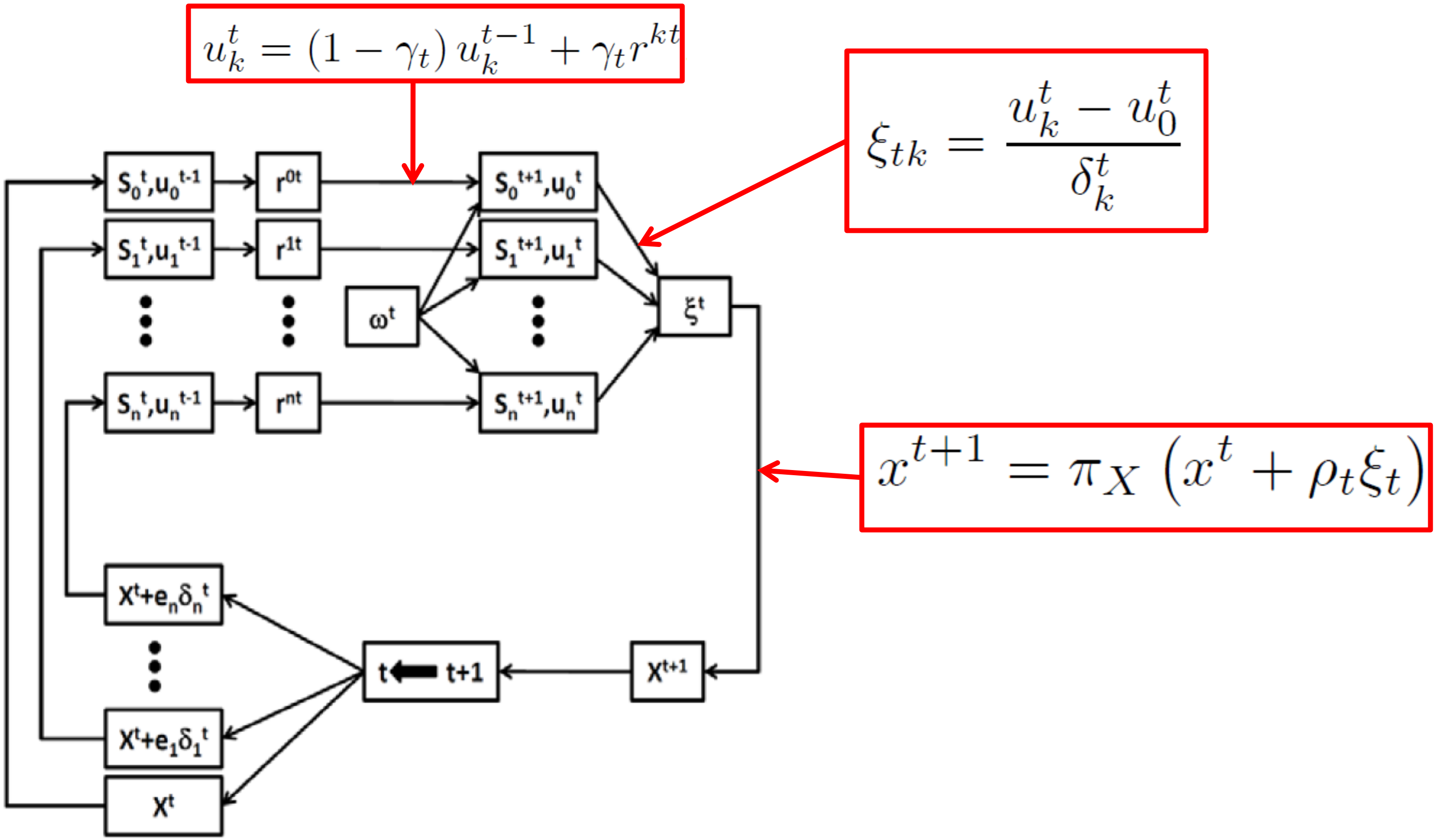
- Transient behavior
- Integration issues between simulation and optimization
- Obvious strategies do not work, for example brute force finite differences
  - At each step:
  - 1. Simulate the system from some initial state for current values of service parameters x^s during time horizon T, obtain the observation of objective $u_{0s}=R^T(x^s,w^s)$.
  - 2. Do the same for the values of x_i^s that differ from x^s in that the i-th variable is incremented by the value δ_s of finite differences, obtain the estimate u_{is}. Do it for all the service parameters.
  - 3. Compute the i-th component of the estimate of the gradient $\xi_i^s=(u_{is}-u_{0s})/\ \delta_s$
  - 4. Perform one step of the SQG method, obtain x^{s+1}
  - 5. Go to step 1.

This takes forever because T should be sufficiently large to get rid of the transient effects.

Integrated simulation and optimization

- Intertwine tightly simulation and optimization: change service parameters according to SQG method *every* simulation step
- Perform $n+1$ parallel simulations for the current point and n shifts with the finite difference step for each of the n parameters using common random numbers
- Utilize previous information in the estimation of function values necessary for the finite differences
- This has an effect of filtering out both noise and transient effects
- Since optimization steps are so lightweight they can be performed by millions in a few minutes on laptop

Integrated simulation and optimization, continued



Optimal values of service parameters are obtained after the end of single simulation run consisting of n+1 simulation threads

Full description of the algorithm

1. Initialization. Select the following parameters of the algorithm:

T *- the number of time periods to perform simulation and optimization;*

$\rho_t \geq 0$ *- the sequence of step sizes for updating of the decision variables* x^t .

γ_t *- the sequence of multipliers for estimation of the objective function* $F(x)$ *from (5).*

θ_t *- the sequence of moving average multipliers for estimation of the solution of problem (5) from the iteration points.*

δ_k^t *- the sequence of finite difference steps for approximating of the gradient of objective function,* $k = 1 : n$.

x^1 *- the starting point for the optimization algorithm.*

$S_k^1 = S^1$ *- the initial state for the simulation of the social network,* $k = 0 : n$.

Select the initial approximation $\tilde{x}^1 = x^1$ *to the solution of the problem (5), the initial estimates* $u_k^0 = u^0$ *of the value of* $F(x^1)$ *and the initial estimate* $\tilde{F}^1$ *to the optimal value of the problem.*

2. Generic step. Suppose that by the start of iteration $t = 1, ..., T$ *the* $k+1$ *simulation processes have arrived at the states* $S_k^t, k = 0 : n$*, the optimization algorithm has generated the value* x^t *of decision variables and* $k+1$ *estimation processes obtained the estimates* u_k^{t-1} *for the values of the objective function* $F(x)$ *at point* x^t *for* $k = 0$ *and points* $x^t + e_k\delta_k^t$ *for* $k = 1 : n$*. On iteration* t *the states* S_k^t*, the estimates* u_k^{t-1} *and point* x^s *are updated as follows.*

2a. The observations $r^{kt}, k = 0 : n$ *of the one period performance measure are obtained (for example, one period revenue from (4)). The observation* r^{kt} *is made from the state* S_k^t *using the values* x^t *of decision variables for* $k = 0$ *and* $x^t + e_k\delta_k^t$ *for* $k = 1 : n$.

2b. The observations ω^t *of random variables* ω *are generated.*

2c. The next states $S_k^{t+1}, k = 0 : n$ *of the social network are obtained using the observations* ω^t*. The state* S_0^{t+1} *is generated starting from the state* S_0^t *according to the network description (1)-(3) and using the value* x^t *of decision variables and the states* $S_k^{t+1}, k = 1 : n$ *are generated starting from the states* S_k^t *and using the values* $x^t + e_k\delta_k^t$ *of decision variables.*

Alexei.Gaivoronski@iot.ntnu.no

Full description of the algorithm, continued

2d. The estimates u_k^t of the objective function $F(x)$ at points x^t for $k=0$ and $x^t + e_k\delta_k^t$ for $k=1:n$ are computed using the observations r^{kt}:

$$u_k^t = (1-\gamma_t)\, u_k^{t-1} + \gamma_t r^{kt},\ k = 0:n$$

2e. The finite difference approximation ξ^t to the gradient of function $F(x^t)$ is computed as follows:

$$\xi_t = (\xi_{t1}, ..., \xi_{tn})\,,\ \xi_{tk} = \frac{u_k^t - u_0^t}{\delta_k^t}$$

2f. The new value x^{t+1} of the decision variables is computed as follows:

$$x^{t+1} = \pi_X\left(x^t + \rho_t \xi_t\right)$$

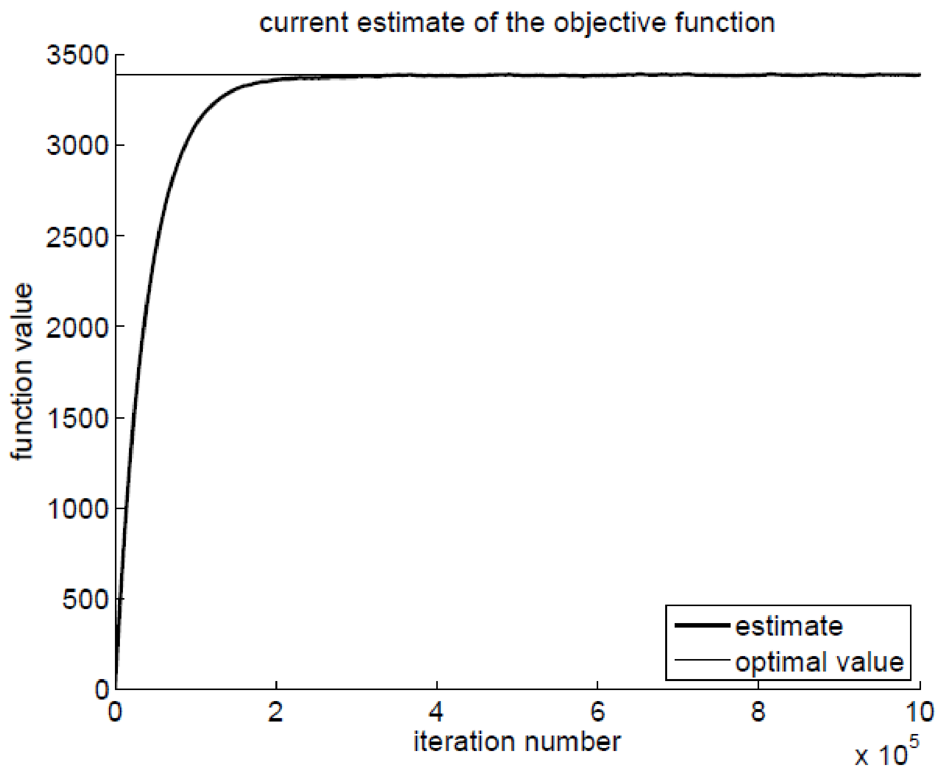
2g. The current approximations to the optimal values of decision variables x and the current approximation $\tilde{F}^{t+1}$ to the optimal value of the problem are computed as the moving average of the iteration points as follows:

$$\tilde{x}^{t+1} = (1-\theta_t)\,\tilde{x}^t + \theta_t x^{t+1}$$
$$\tilde{F}^{t+1} = (1-\theta_t)\,\tilde{F}^t + \theta_t x^{t+1}$$

2h. Stop if $t = T$ and take the average of the last T_1 values of $\tilde{x}^t$ and $\tilde{F}^t$ as the final approximations $\tilde{x}$ and $\tilde{F}$ to the optimal solution of the problem(5) and its optimal value.□

Tuning of the algorithm on the problem with known solution

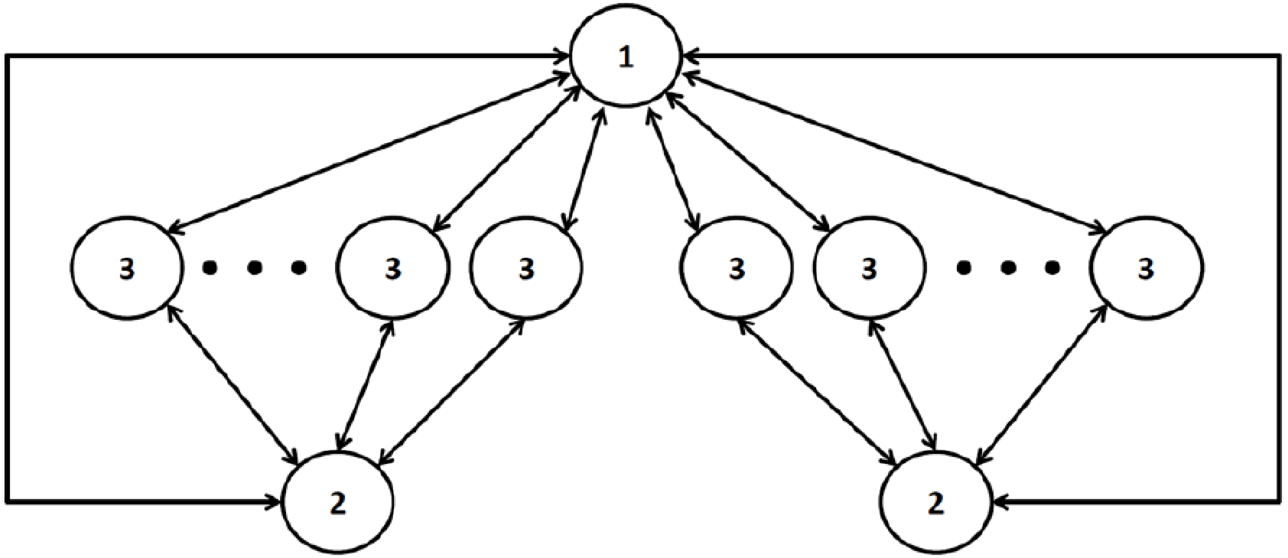
- If the network has fixed structure the process is described by Markov chain with 2^N states. For small N one can find stationary probabilities of the chain and find the optimal prices using off-shelf NLP software. This is used as a benchmark



NTNU
Alexei.Gaivoronski@iot.ntnu.no

Study of networks with different structure

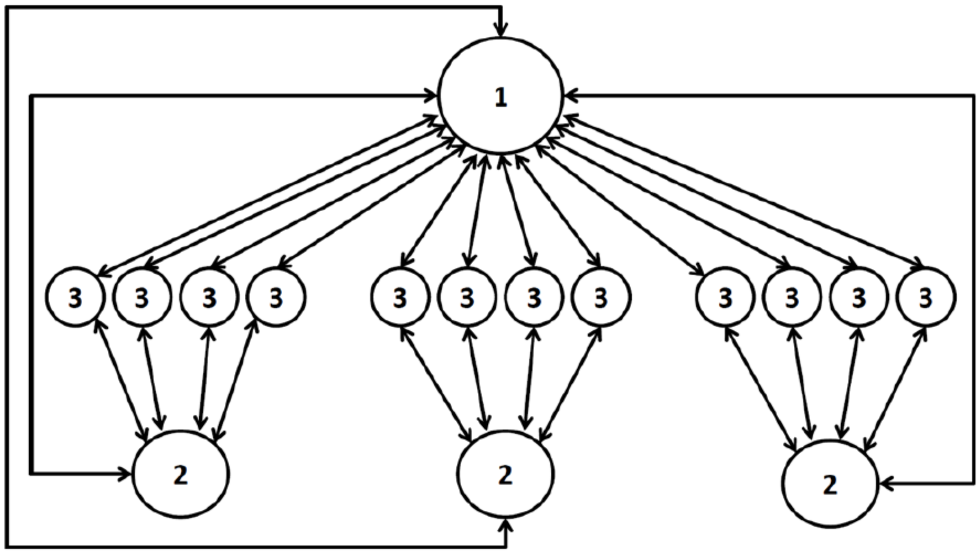
- Network of type A



- Global leader 1 connected with everybody
- Local leaders 2 connected with half of the rest and the leader
- Ordinary nodes 3 connected only with leader1 and one of the local leaders 2

Study of networks with different structure, continued

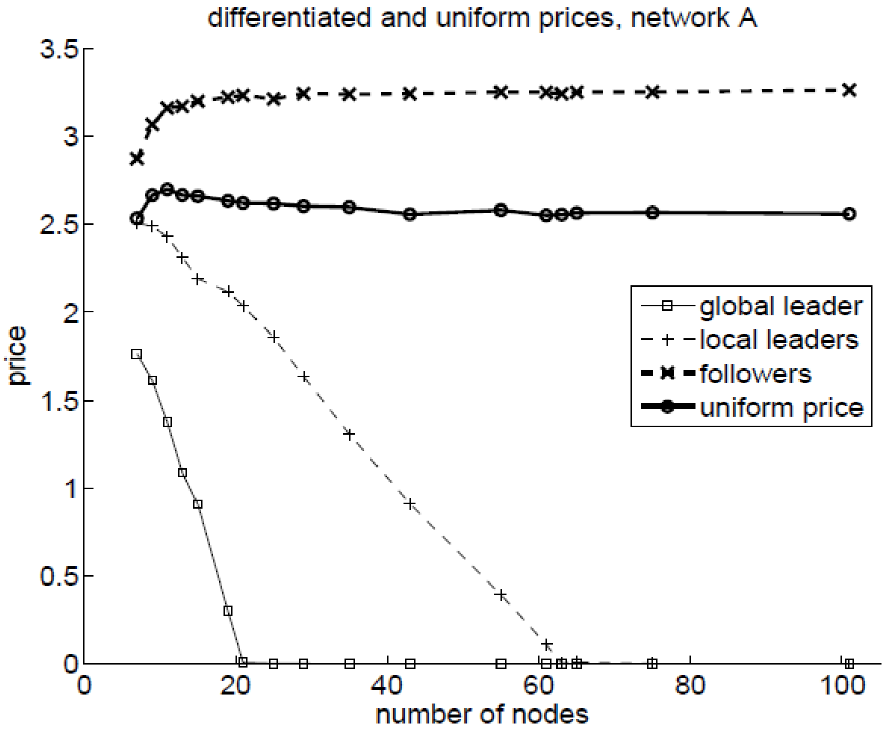
- Network of type B

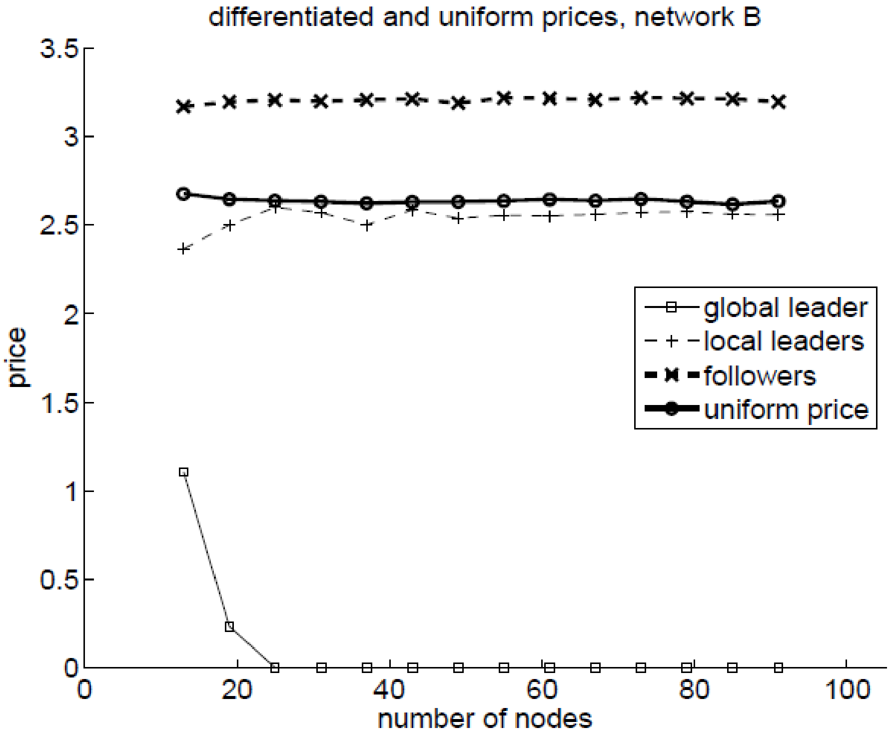


- Global leader 1 connected with everybody
- *L* local leaders 2 connected with *M* followers and the global leader
- Followers 3 connected with global leader 1 and one of the local leaders 2

Optimal prices, networks with fixed structure

- Differentiation does matter

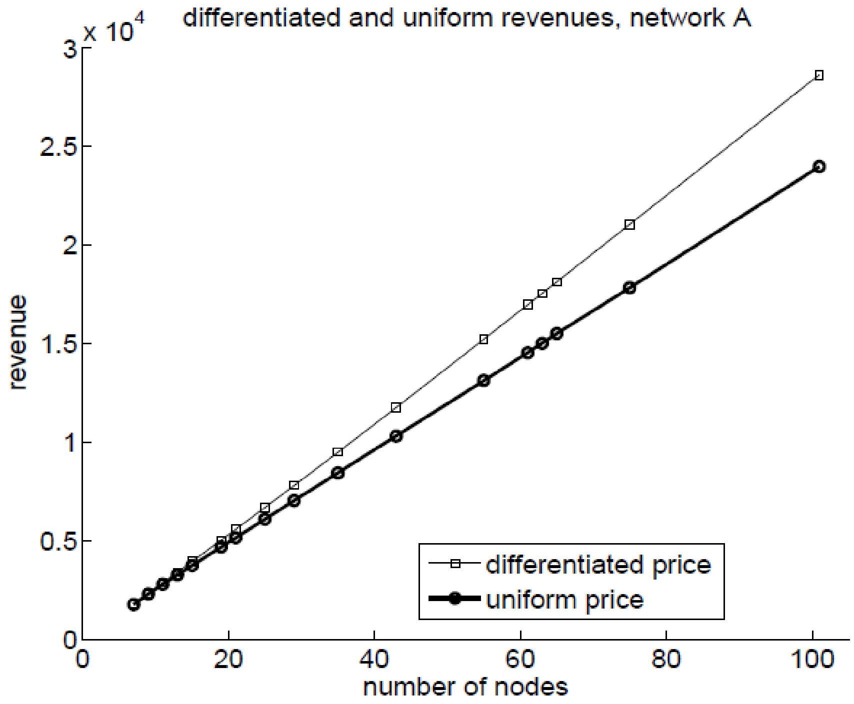


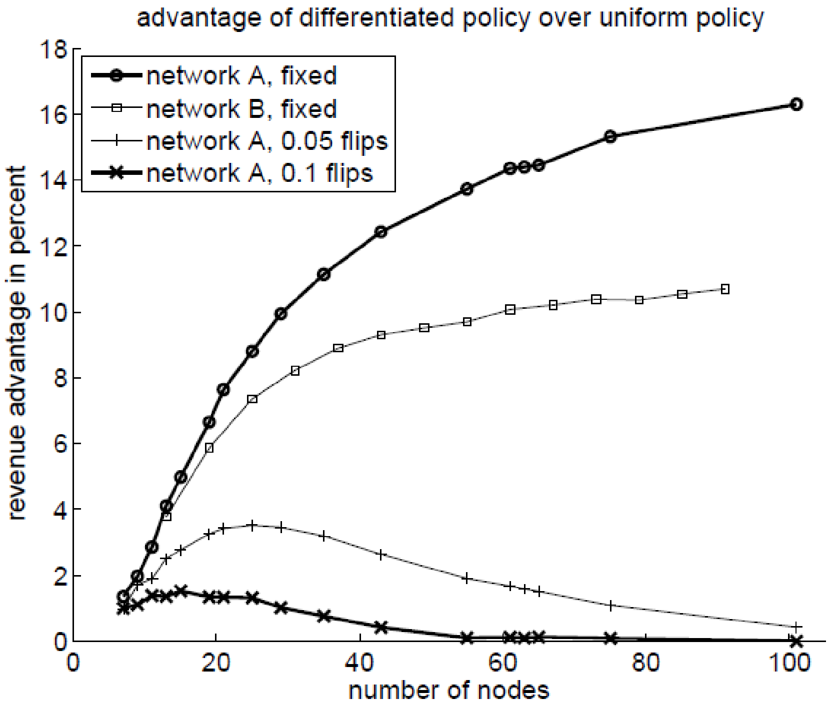


Alexei.Gaivoronski@iot.ntnu.no

Optimal revenues, networks with fixed structure

- Differentiation does matter, but random errors in connections can destroy the picture

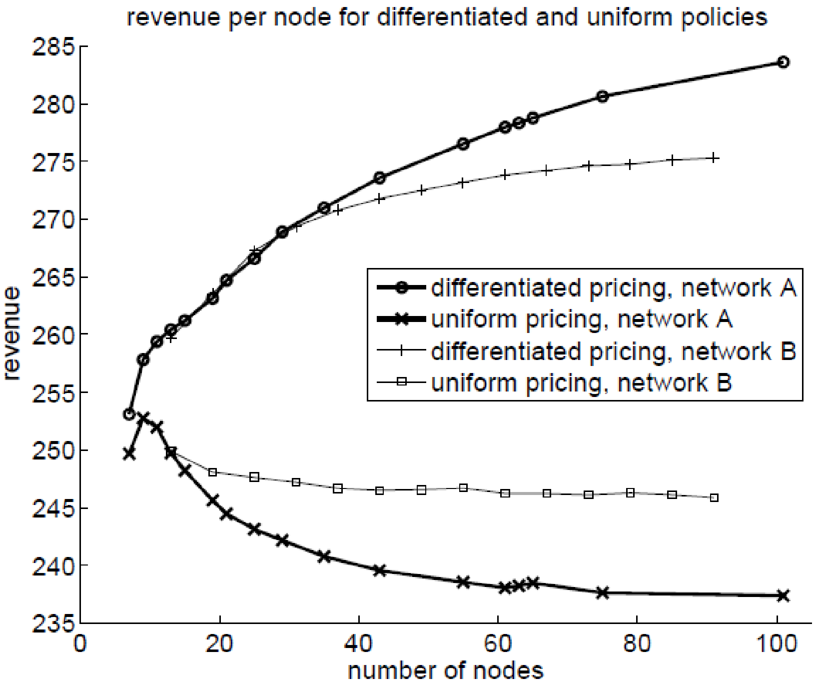


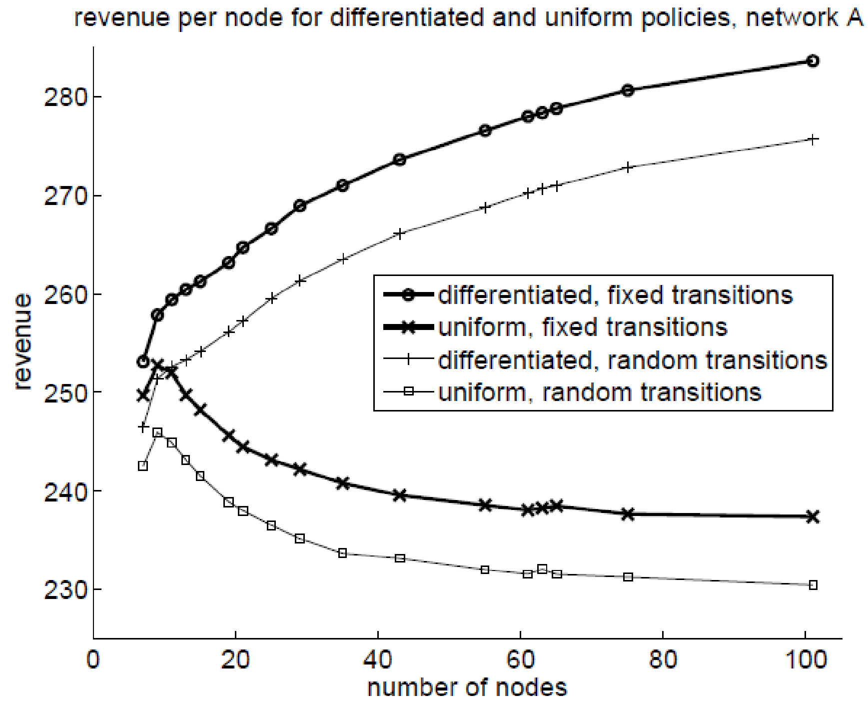


Alexei.Gaivoronski@iot.ntnu.no

Optimal revenues, networks with fixed structure

- Differentiation does matter, errors in transition probabilities do not matter much

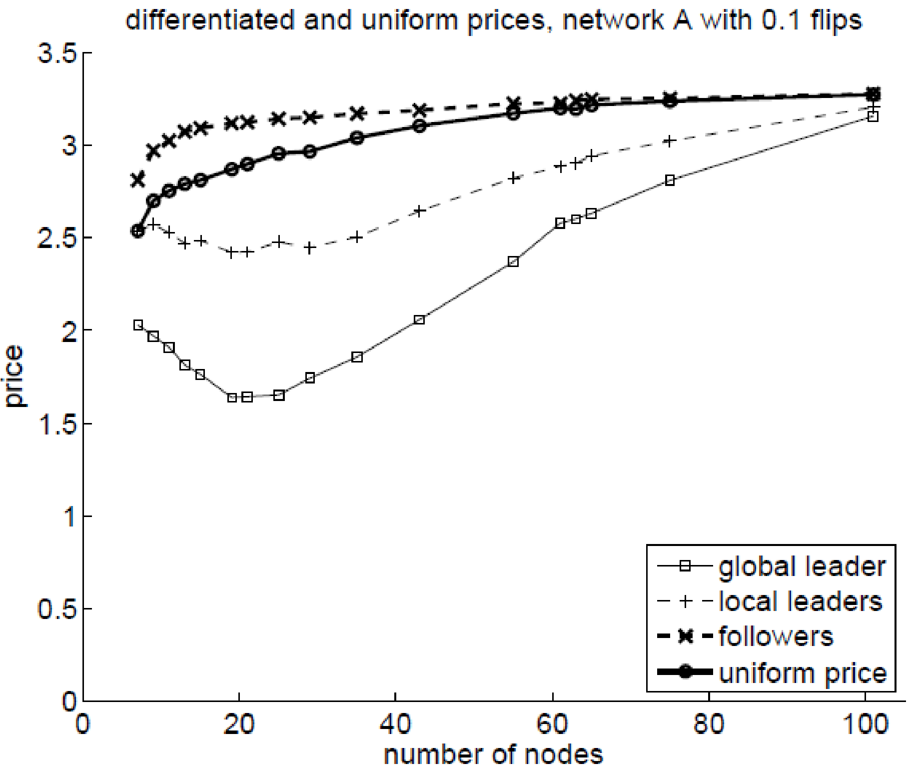


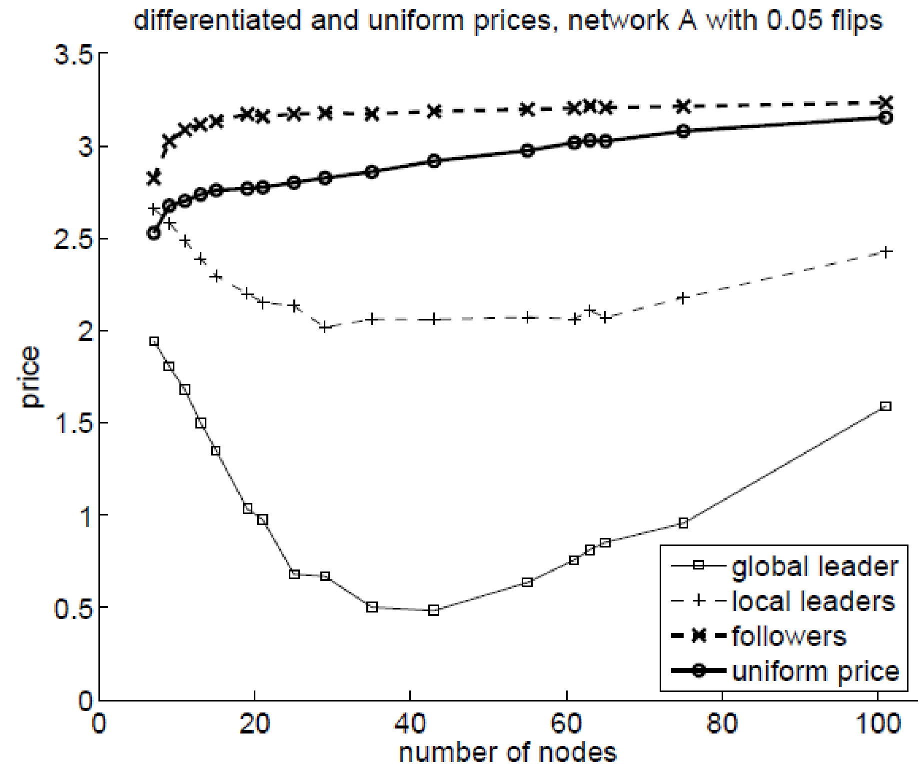


Alexei.Gaivoronski@iot.ntnu.no

Optimal prices, networks with randomly changing structure

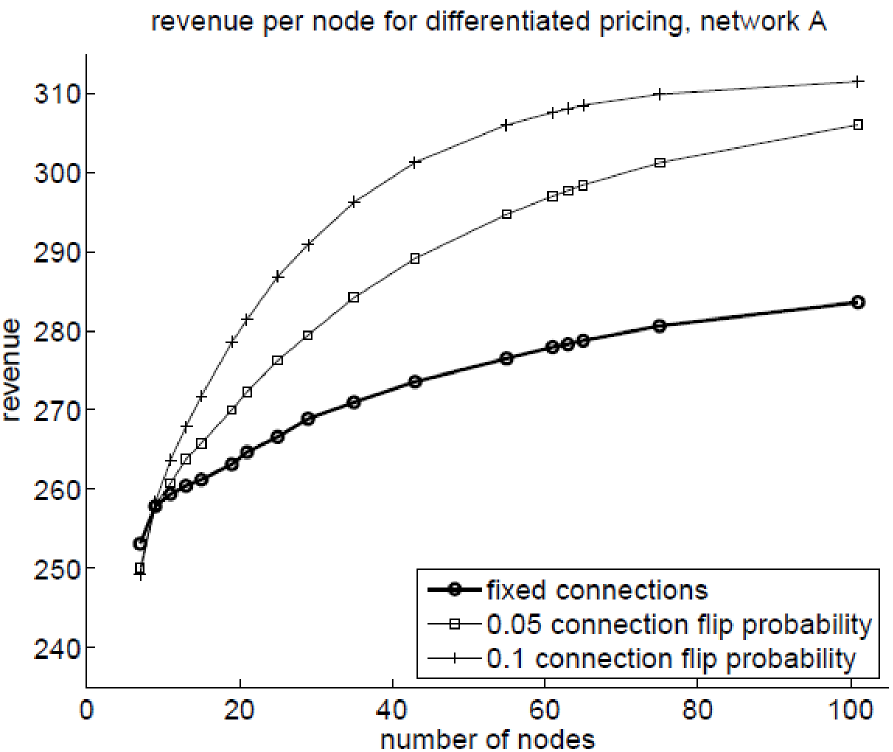
- If connections flip randomly the prices converge

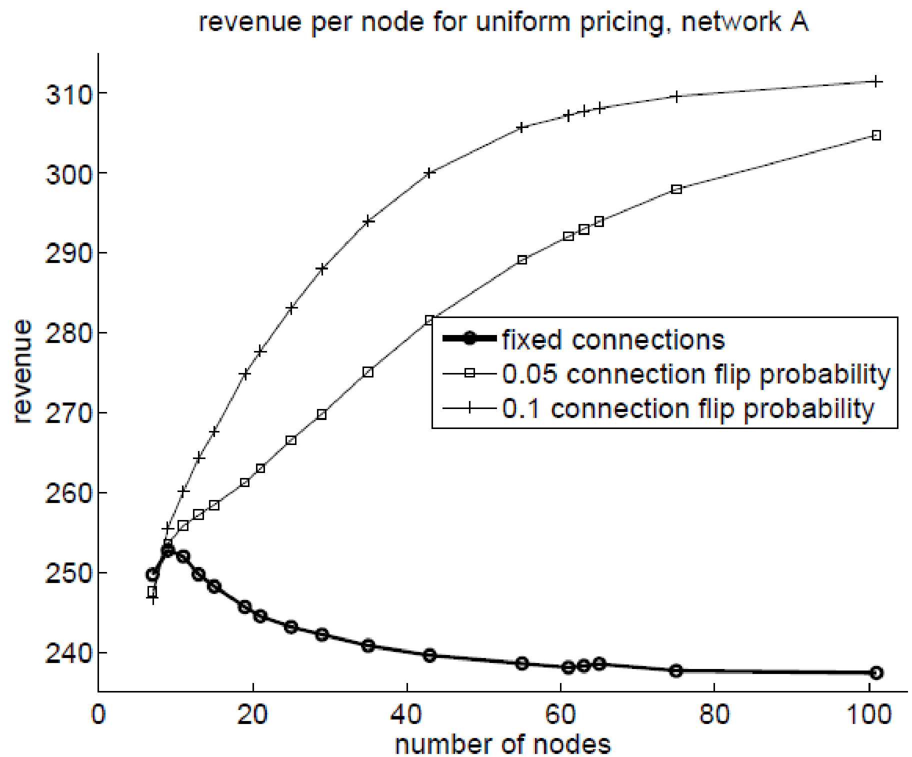




Optimal revenues, networks with randomly changing structure

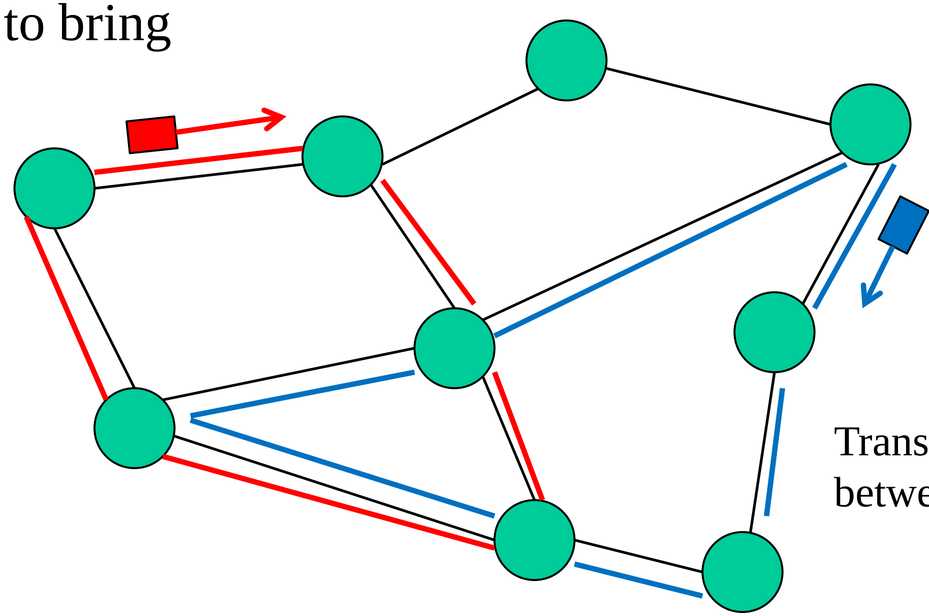
- If connections flip randomly the revenues grow due to increased network effects





Example 2: Transportation between supply/demand nodes with inventories

- Set of nodes connected with links (roads, waterways …) with supply/demand (or both) for a collection of items
- Transportation (trucks, ships) operates on links that takes items in excess and bring them where they are lacking
- Transportation time, finite inventories, costs of transportation, inventory, backlog
- Decisions taken before one step before inventory is known: where to take and where to bring



Transportation of empty containers between ports

Comparison of scenario tree deterministic equivalent with optimization/simulation

$$\min_{u^r_{ik},w^r_{ik},x^r_{jk}y^r_{jk},z^r_{ik}} \sum_{r=1}^{R} p_r \sum_{j=1}^{J}\sum_{k=1}^{K}\left(a^r_{jk}x^r_{jk}+b^r_{jk}y^r_{jk}\right)+\sum_{r=0}^{R'}\sum_{m\in\Omega_r} p_m \sum_{i=1}^{I}\sum_{k=1}^{K}\left(g^m_{l_{im}k}u^r_{ik}+h^m_{l_{im}k}z^r_{ik}\right)$$

subject to constraints

$$x^m_{jk}-y^m_{jk}+\sum_{i:j=l_{im}} u^0_{ik}-\sum_{i:j=l_{im}} z^0_{ik}=x^0_{jk}+d^{0m}_{jk},\ j=1:J,\ k=1:K,\ m\in\Omega_0$$

$$x^m_{jk}-x^r_{jk}-y^m_{jk}+\sum_{i:j=l_{im}} u^r_{ik}-\sum_{i:j=l_{im}} z^r_{ik}=d^{mr}_{jk},\ j=1:J,\ k=1:K,\ r=1:R',\ m\in\Omega_r$$

$$w^m_{ik}-u^0_{ik}+z^0_{ik}=w^0_{ik},\ i=1:I,\ k=1:K,\ m\in\Omega_0$$

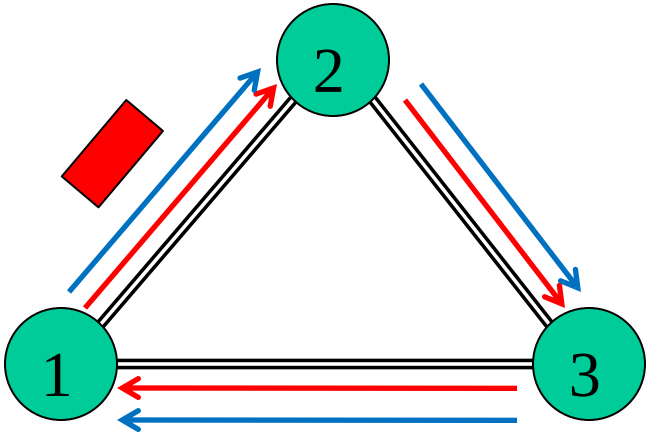
$$w^m_{ik}-w^r_{ik}-u^r_{ik}+z^r_{ik}=0,\ i=1:I,\ k=1:K,\ t=2:T,\ r=1:R',\ m\in\Omega_r$$

$$\sum_{k=1}^{K} C^c_k w^r_{ik}\le C^s_i-v^0_i,\ i=1:I,\ r=1:R$$

Alexei.Gaivoronski@iot.ntnu.no

A small problem due to huge dimension of deterministic equivalent

- 3 ports, 1 type of conatiners,1 ship that travels in one day between ports, 7 days, forecasts of demand/supply are known, normal distribution around forecasts, independent forecast errors

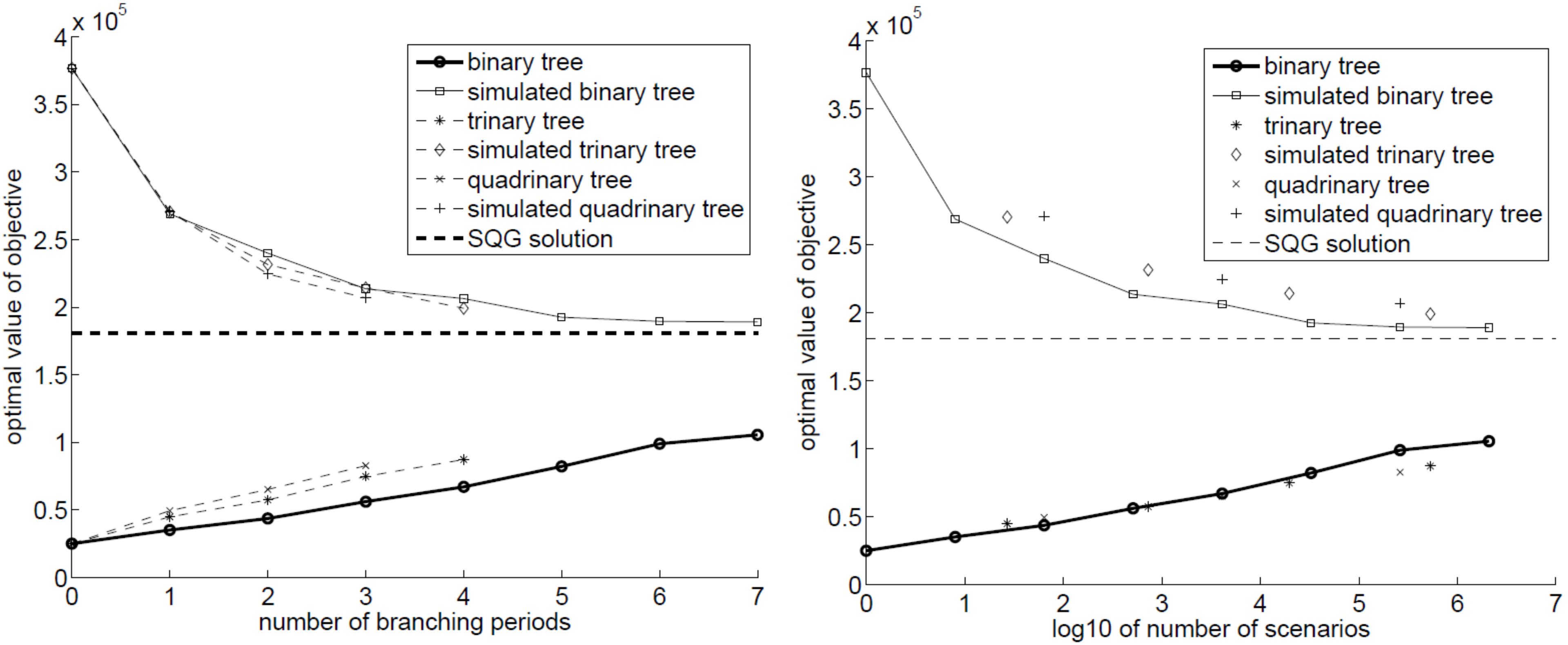


- Binary tree 7 periods on each of 3 demands, 7 days: $2.1*10^6$ scenarios, $1.7*10^7$ variables, $1.2*10^7$ constraints, CPLEX 12.1, 8 hours of OPL 6.3 time of single core 2.93 GHz, 8 GB of memory, 23 GB virtual memory allocated
- Trinary tree first 4 periods: $5.3*10^5$ scenarios, $1.8*10^7$ variables, $1.1*10^7$ constraints, 34 hours of OPL 6.3 time
- What if we use the most powerful cloud available? Trinary tree first 5 periods $1.4*10^7$ scenarios, $3.6*10^8$ variables, $2.2*10^8$ constraints

Alexei.Gaivoronski@iot.ntnu.no

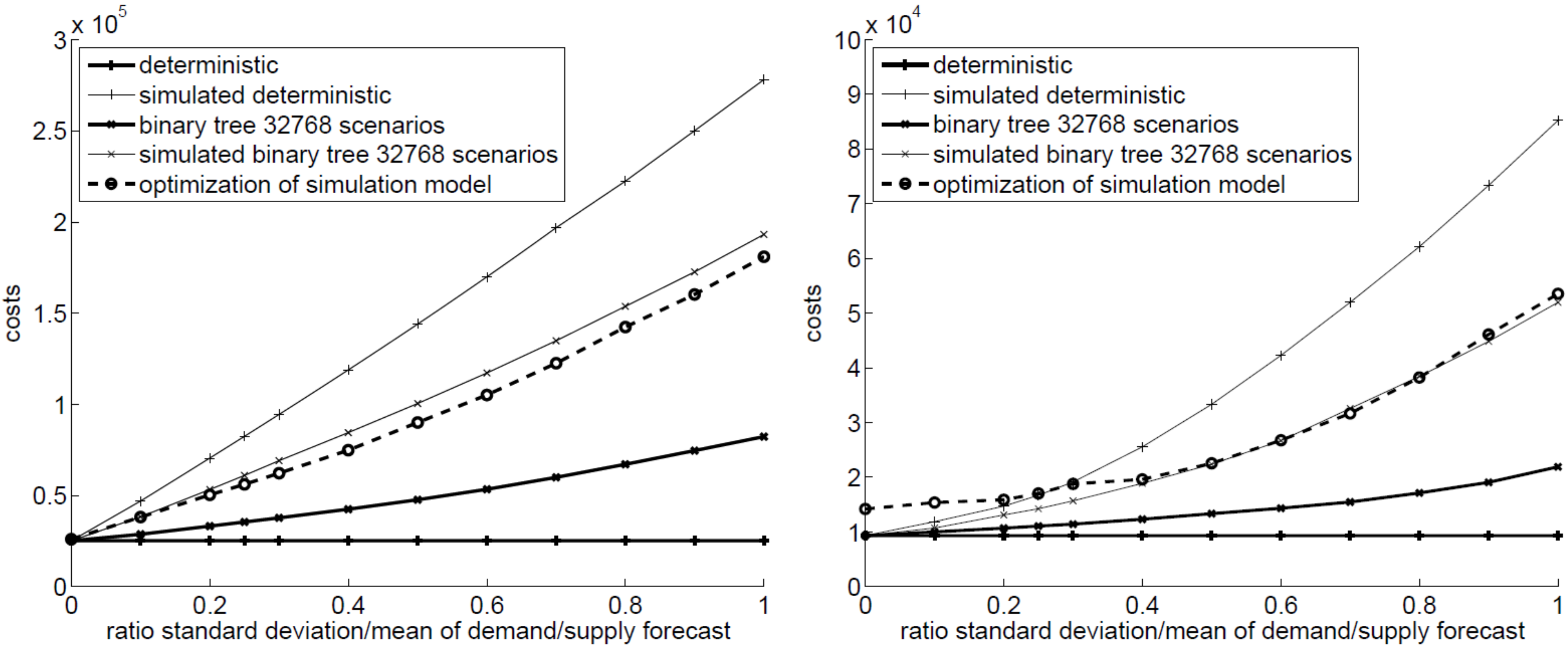
Which tree to choose

- Professor Pflug has proposed how to do this optimally



- Conclusion: binary tree is superior to other trees for the same number of scenarios

Comparison for different levels of uncertainty



Conclusion, part 1

- Optimization of simulation models is important methodology for solving complex optimization problems under uncertainty with growing relevance for applications
- Algorithmic development is important
- Contribution of Professor Pflug provides theoretical and algorithmic underpinning to this methodology

Conclusion, part 2

- See next slide


Best wishes
to Georg!