Dynamic modelling and network optimization

Risto Lahdelma
Aalto University
Energy Technology
Otakaari 4, 02150 Espoo, Finland
risto.lahdelma@aalto.fi

Outline

- Dynamic systems
  - Dynamic energy models
  - Specialized dynamic programming (DP) algorithm
- Network flow modelling

Dynamic systems

- A dynamic system is one which develops in time
  - Opposite: static system
- Normally, a dynamic system is modelled by discretizing it into a sequence static models that are connected by dynamic constraints

Dynamic energy models

- Dynamic constraints result from
  - Energy storages
  - Transient constraints (limits for rate of increasing/decreasing production)
  - Startup and shutdown costs
  - Startup and shutdown constraints
- Examples:
  - Yearly CHP planning model represented as a sequence of 8760 hourly models
  - Daily hydro power scheduling represented as a sequence of 96 15min models

Dynamic optimization

- Different ways to model and solve dynamic systems exist
- Multiperiod LP/MILP models
  - Network flow model (LP)
- General mathematical optimization models
- Heuristic techniques (do not guarantee optimality)
- **Dynamic programming algorithm**

Dynamic programming (DP)

- Special optimization method for certain kind of dynamic optimization problems
  - Network based modelling technique
  - Cost function must be additive with respect to periods
    - Each time step adds some (positive) amount to the cost function
  - Static models at nodes **do not need be linear**

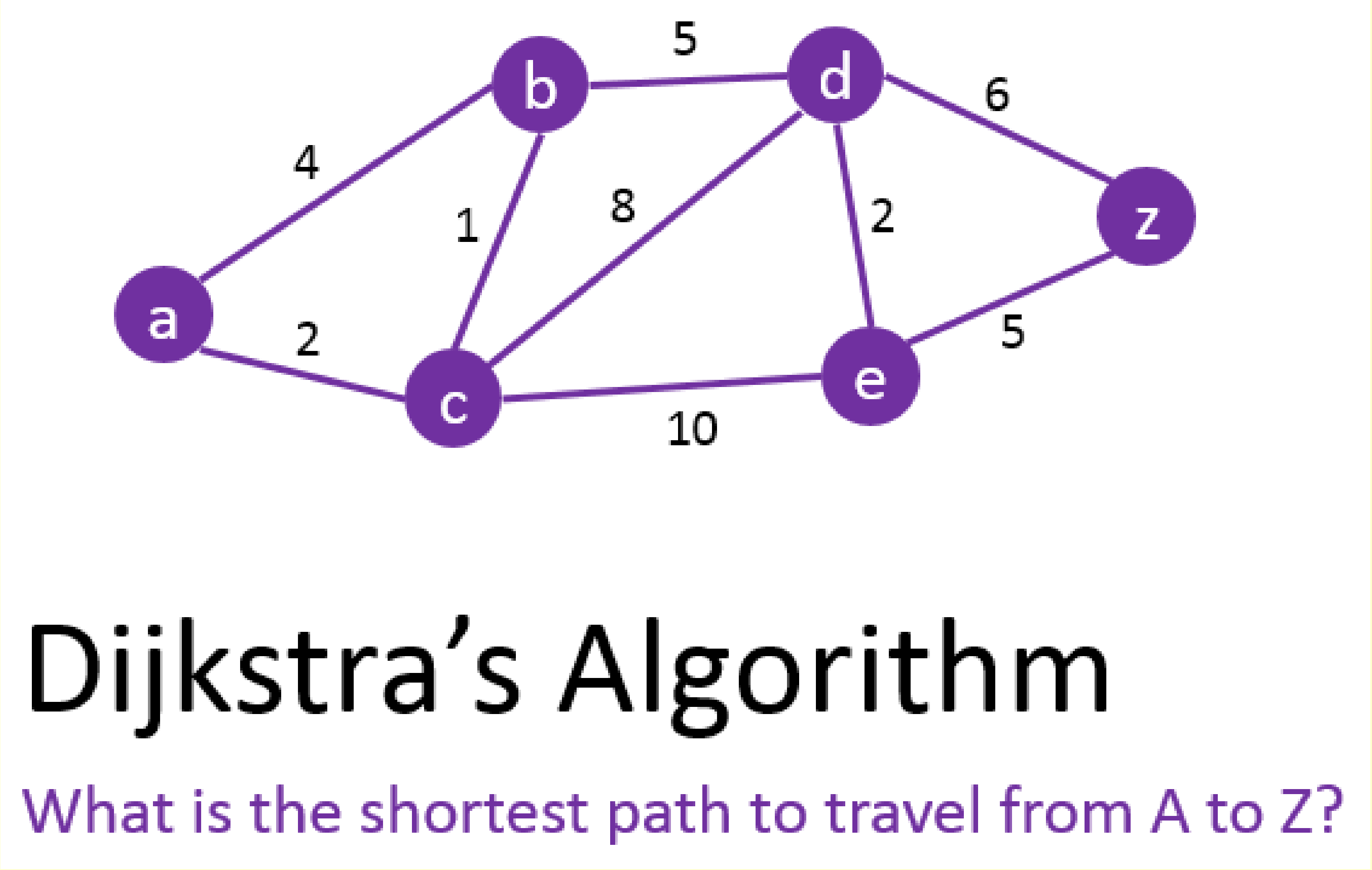
Dynamic programming (DP)

- Based on **Dijkstra's graph search algorithm**
  - in a graph with non-negative edge path costs the algorithm determines the shortest path between a given node and all other nodes
  - Given node can be start or end node
- Optimality principle:
  - If the shortest path from node **s** to **t** goes through node **b** (**s**,…,**b**,…,**t**), then
    subpath (**b**,…,**t**) is the shortest path between **b** and **t**;
    also subpath (**s**,…,**b**) is the shortest path between **s** and **b.**

Dijkstra's graph search algorithm

- Used for finding shortest path in a graph
- Shortest can mean
  - Shortest distance, shortest travel time, cheapest way to travel, or minimization of any other additive measure
  - Only required that each arc adds up to the objective
    - Arc costs may not be negative – negative costs may result in infinitely good cyclic paths
- Used e.g. in GPS navigators to find shortest or fastest route
- The algorithm has almost linear time complexity with respect to number of nodes & arcs
  - Extremely fast even for huge networks!

Dijkstra’s graph search algorithm

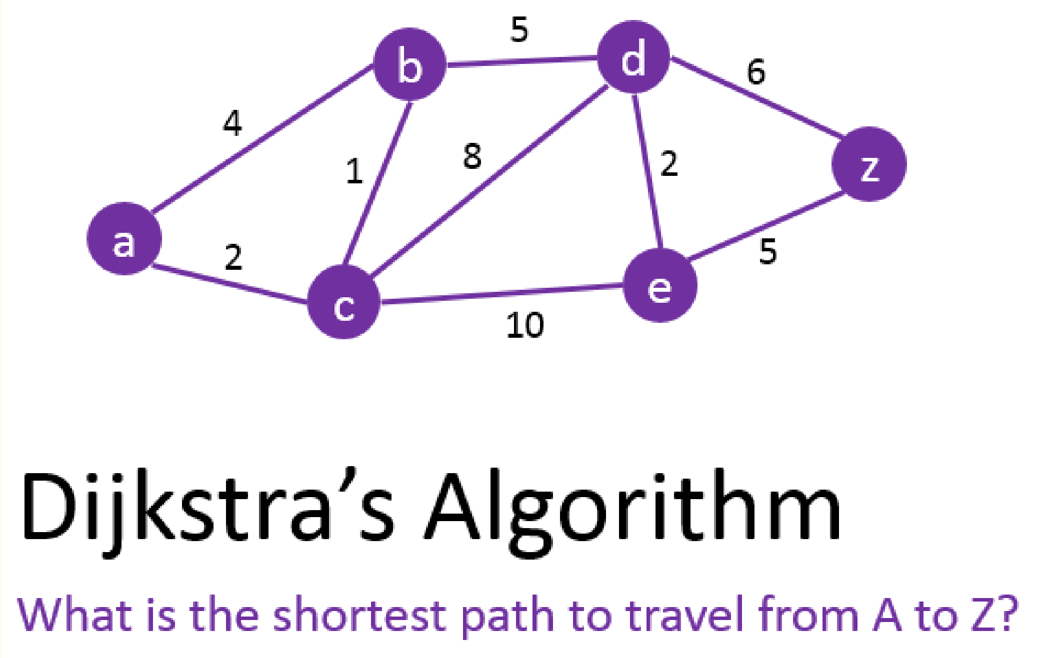


Dijkstra's graph search algorithm

- The algorithm maintains a set of paths **L** such that
  - each path (**s**,…,**b**)$\in$**L** is the shortest so far found path to **b**, but it is not yet known if this path is optimal.
  - Initially **L** includes only node **s** for which a zero-length path (**s**) is known
- While **L** is non-empty, the algorithm works iteratively:
  - The *shortest* of the so far known paths (**s**,…,**b**) must be *optimal*, because reaching **b** by extending some of the other paths would yield a longer path (non-negative costs add up)
  - Path (**s**,…,**b**) is removed from **L**
  - Each node **c** adjacent to **b** is explored to see if path (**s**,…,**b**,**c**) is a shorter path to **c** than the so far shortest found path.
  - **L** is updated: shorter paths are inserted, longer paths removed.
- With proper datastructures the algorithm has almost linear time complexity with respect to number of nodes & arcs!

Dijkstra's graph search algorithm

1. L={(a):0}
2. L={(a,c):2, (a,b):4)}
3. L={(a,c,b):3, ~~(a,b):4,~~ (a,c,d):10, (a,c,e):12}
4. L={(a,c,b,d):8, ~~(a,c,d):10,~~ (a,c,e):12}
5. L={(a,c,b,d,e):10, ~~(a,c,e):12,~~ (a,c,b,d,z):14}
6. L={(a,c,b,d,z):14}
7. L={}



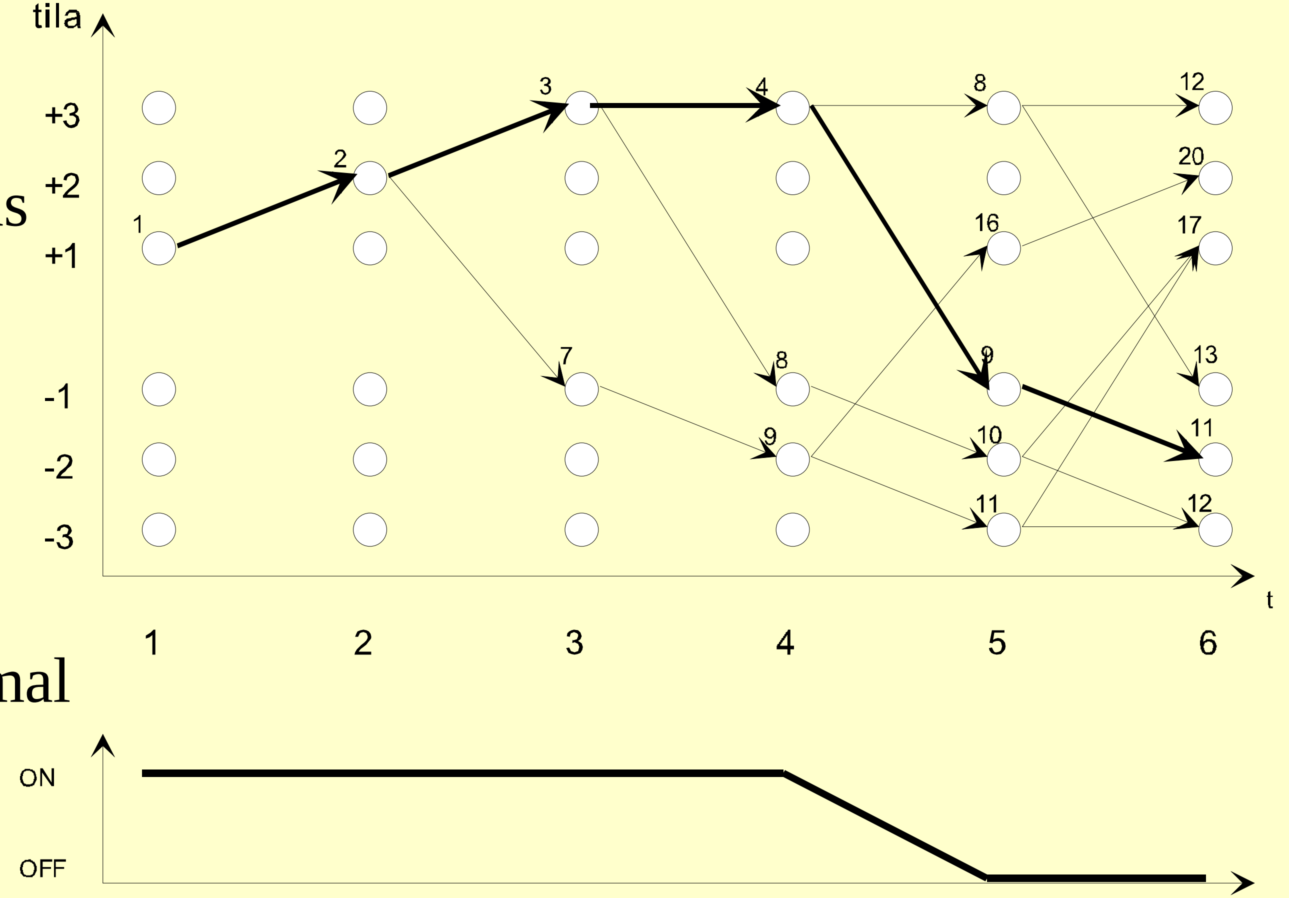
- At each step, first element of **L** is the shortest (optimal) path from a to corresponding end node
- Shortest path a to z is found at step 6: (a,c,b,d,z) with cost 14

Unit commitment: Production plant startup/shutdown scheduling

- Unit commitment of power plants means determining for each time step which *units (*plants) should be up and running, and which should be switched off
- Unit commitment results in a complex MILP problem, because
  - Binary ON/OFF status variables for each plant and time step result in a large number of binary variables
  - Time periods are linked together by startup and shutdown costs & constraints
  - Dynamic storage constraints may also be required

Dynamic programming (DP) for unit commitment of a single unit

- Status of the unit is an integer +1,+2,+3… or -1,-2,-3…, indicating for how many periods the plant has been on or off
- The arcs denote allowed transitions with associated costs
- The shortest path from the initial state to the final period is the optimal way to operate



DP for production plant startup/shutdown scheduling

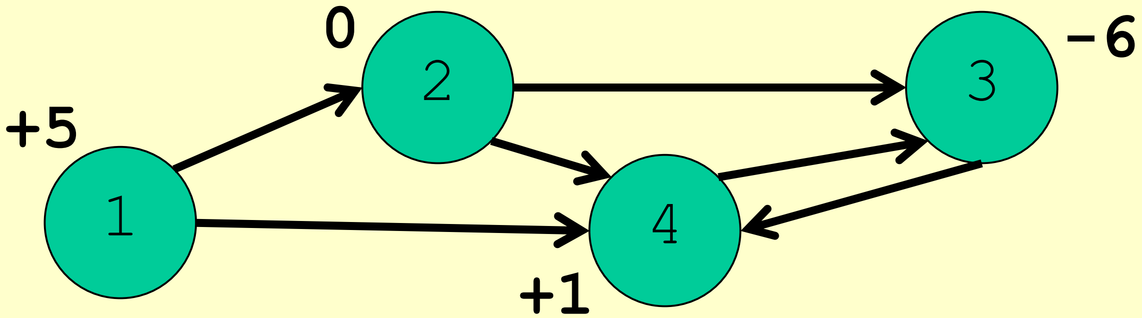
- Arcs for illegal transitions are omitted
  - When startup/shutdown is not allowed (minimum on/off time)
  - When a plant is required to be running or off
    - Scheduled service break
- The costs for allowed transitions include
  - Startup/shutdown costs
  - Operating costs within period in the given on/off state subtracted by possible revenues
- Operating costs can be determined by optimizing static model for the period
  - Arbitrary optimization model can be used

Limitations of DP for unit commitment

- With one plant, the number of on/off states per period is reasonable (6 in previous example)
- With multiple plants, it is necessary to consider all combinations of on/off-states of all plants
  - With two plants 36 combinations: (+1,+1), (+1,+2), (+1,+3), (+1,-1), (+1,-2), (+1,-3), (+2,+1,) ...
  - Problem size and solution time grows exponentially by the number of plants
  - Solution: Sequential DP (see article by Rong, Hakonen, Lahdelma)
- Including storages further complicates the model
  - Storage level is discretized and treated as a plant

Network flow model

- A **network** consists of **nodes** and connecting **arcs**
  - Some commodity (power, heat, …) can flow through the arcs
  - Normally arcs are **directed** allowing flow in only one direction
  - Two-way flow is represented as a pair of opposite arcs



- **Attributes** are associated to nodes and/or arcs
  - **Supply**/demand $\pm d_j$ of commodity at each node
  - Transfer **price** c_{ij} through each arc
  - Possibly a maximum **capacity** u_{ij} for each arc

The transshipment network flow model

- The aim is to determine flows x_{ij} through each arc so that
  - All nodes in the network are balanced
    - For this to succeed, it is necessary that total supplies/demands at nodes $\Sigma_i\, d_i = 0$
  - Overall transportation costs are minimized
  - Transshipment = commodity can pass through other nodes before reaching its final goal

The transshipment network flow model

- LP-formulation

  Min $\Sigma_i \Sigma_j c_{ij} x_{ij}$ // minimize total costs

  s.t.

  $\Sigma_j x_{ij} - \Sigma_j x_{ji} = d_i$ for each node i

  $x_{ij} \geq 0$ for each arc (i,j)

  (optionally also capacity constraints $x_{ij} \leq u_{ij}$)

  – Here
    - x_{ij} is flow from node i to j
    - c_{ij} is unit cost for flow from node i to j
    - d_i is supply at node i, negative value = demand
    - u_{ij} is maximum allowed flow from node i to j

The transshipment network flow model

- It is a special case of an LP model
- Can be solved using
  - generic Simplex algorithm for LP
  - much more efficient network simplex algorithm
- Applies to a wide variety of different problems
  - Static models
    - Power production and transfer between market areas
    - District heat production and transmission
  - Dynamic models
    - Hydro power optimization

Transshipment flow problem example: Power transmission problem

- Minimize transmission costs while balancing fixed supply & demand in different areas
- Decision variables

  x_{ij} power transmission from area i to j (MWh)
- Parameters

  c_{ij} power transmission cost from area *i* to *j* (€/MWh)

  u_{ij} capacity limit for transmission from area *i* to *j* (MWh)

  d_i net supply (supply – demand) in area *i*
- Model

  Min $\Sigma_i \Sigma_j c_{ij} x_{ij}$

  s.t. $\Sigma_j x_{ij} - \Sigma_j x_{ji} = d_i$ for each area i

  $0 \leq x_{ij} \leq u_{ij}$

Numerical example: Power transmission problem

- Areas 1, … 4
- Supply/demand, transmission costs, infinite capacities

$$\mathbf{d} = \begin{bmatrix} 5 \\ 0 \\ -6 \\ 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} - & 1 & \infty & 3 \\ \infty & - & 2 & 1 \\ \infty & \infty & - & 1 \\ \infty & \infty & 1 & - \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} 0 & \infty & 0 & \infty \\ 0 & 0 & \infty & \infty \\ 0 & 0 & 0 & \infty \\ 0 & 0 & \infty & 0 \end{bmatrix}$$

Min x12 + 3*x14 + 2*x23 + x24 + x34 + x43

s.t.

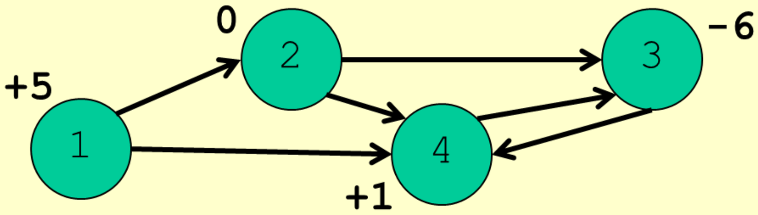
x12 + x14 = 5; // area 1

x23 + x24 – x12 = 0; // area 2

x34 – x23 – x43 = –6; // area 3

x43 – x14 – x24 – x34 = 1; // area 4

x12, x14, x23, x24, x34, x43 >= 0;



Numerical example: Power transmission problem

- In matrix format (min **cx** s.t. A**x** = **b**, **x**≥0) we have

$$\mathbf{x} = \begin{bmatrix} x_{12} & x_{14} & x_{23} & x_{24} & x_{34} & x_{41} \end{bmatrix}^T$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & & & & \\ -1 & & 1 & 1 & & \\ & & -1 & & 1 & -1 \\ & -1 & & -1 & -1 & 1 \end{bmatrix}$$

- **A** has +1 and **-**1 on each column
- One of the constraints is redundant
  - Any row of A is the negated sum of all other rows

Solution of transshipment flow problem

- Special **Network Simplex** algorithm can solve the transshipment problem even faster than generic LP
- Assuming that none of the transmission costs is negative, the **optimal solution will not contain cyclic flows**
  - A cyclic flow means that some amount of commodity completes a cycle ending at the same node where it started
- Assuming that the network is connected, the optimal solution will form a *spanning tree* of the network
  - A **tree** is a connected *acyclic* (cycle-free) graph
  - **Spanning tree** is a tree that connects all nodes of the graph
  - To find optimum, it is sufficient to examine only tree-solutions
  - Tree-solutions are equivalent to basic solutions of LP formulation

Piecewise linear approximation of power plant with convex characteristic

- Assume a power (or heat) plant with convex characteristic (c,p), c = cost, p = power (or heat)
  - Two part piecewise linear approximation of characteristic
  - Model

$\text{Min } c_1{*}p_1 + c_2{*}p_2$

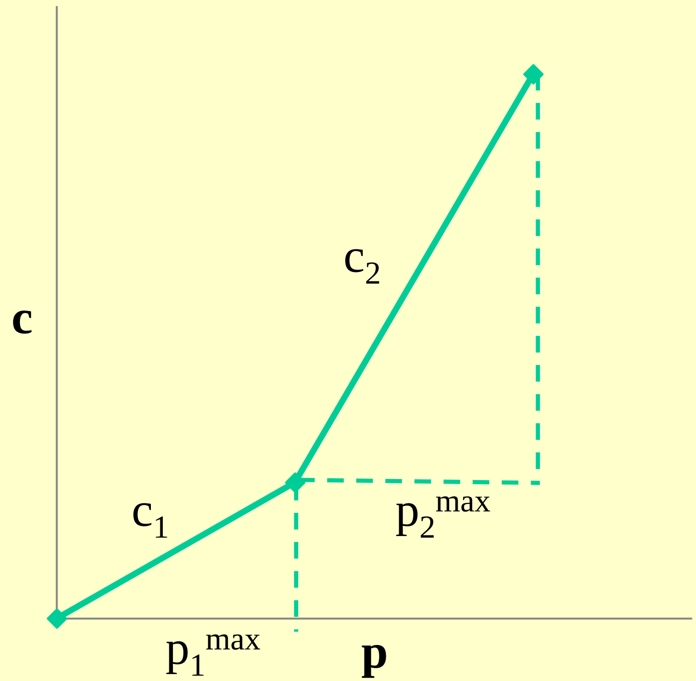
s.t.

$p_1 + p_2 = P$ // fixed demand

$p_1 \le p_1^{max}$

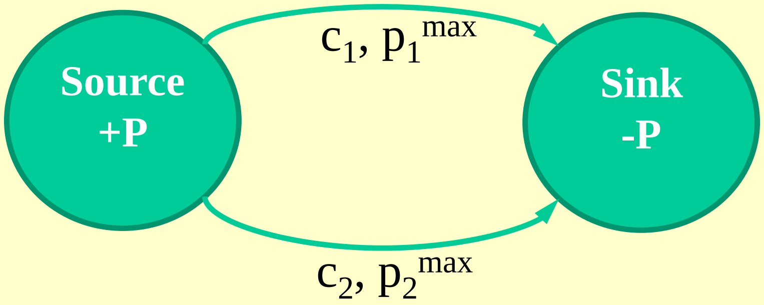
$p_2 \le p_2^{max}$

$p_1, p_2 \ge 0$



Encoding production model as transshipment problem

- Previous power or heat plant with convex characteristic approximated with 2 line segments
  - Source node with supply P, sink node with demand –P
  - For each line segment an arc with cost and capacity
  - Model



Min $c_1 * x_1 + c_2 * x_2$

s.t.

$x_1 + x_2 = P$ // source node

($-x_1 - x_2 = -P$ // sink node)

$x_1 \leq x_1^{max}$

$x_2 \leq x_2^{max}$

$x_1, x_2 \geq 0$

Combined power production and transmission problem

- Problem definition
  - Minimize overall power production costs across multiple market areas (e.g. countries or cities inside one country)
  - Each area has local power production (both CHP and condensing power)
  - Each area has specified demand for power and heat
  - It is possible to transmit power between areas using capacitated transmission lines (but heat cannot be transmitted across areas)
  - The target is to determine how much power should be produced in each area and how power should be transmitted between areas
- For example the Nordic power market (NordPool) ideally implements such production cost minimization

Three condensing power plant model

- We have three power plants:
  - capacities 100, 200 and 300 MWh
  - production costs 25, 30, 22€/MWh
  - Power demand is P
- Define an LP model for minimizing the production costs and solve the problem for power demand P = 50, 150, 250, 350 MWh using Solver

Next, reformulate the problem as a network flow problem

Define an LP model for minimizing the production costs.

- Decision variables

  x1, x2, x3: power production at each plant (MWh)

- Parameters

  P: power demand (MWh)

- Model

  min 25*x1 + 30*x2 + 22*x3

  s.t.

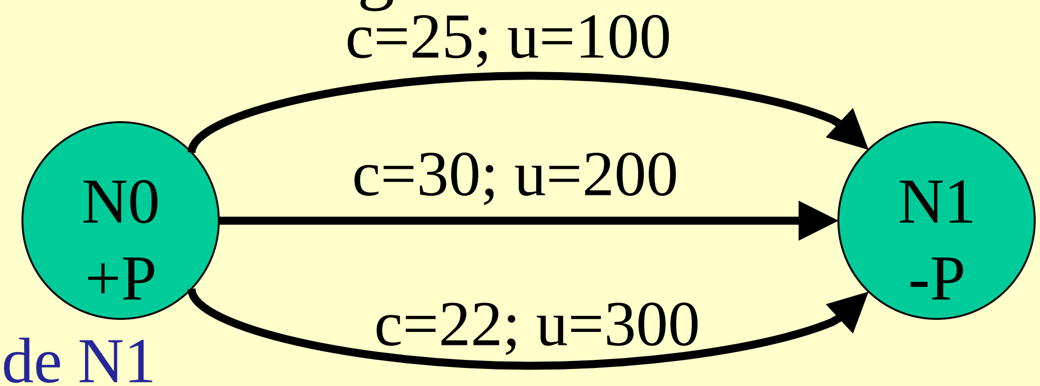
  x1 + x2 + x3 = P;

  x1 <= 100; x2 <= 200; x3 <= 300;

  x1, x2, x3 >= 0;

Reformulate the problem as a network flow problem and draw a picture

- Define two nodes
  - Source N0 with supply +P, sink N1 with demand –P
- Define one arc for each linear segment
- Model



min 25*x1+30*x2+22*x3

s.t.

-x1 - x2 - x3 = -P; // for node N1

x1 + x2 + x3 = P; // for node N0

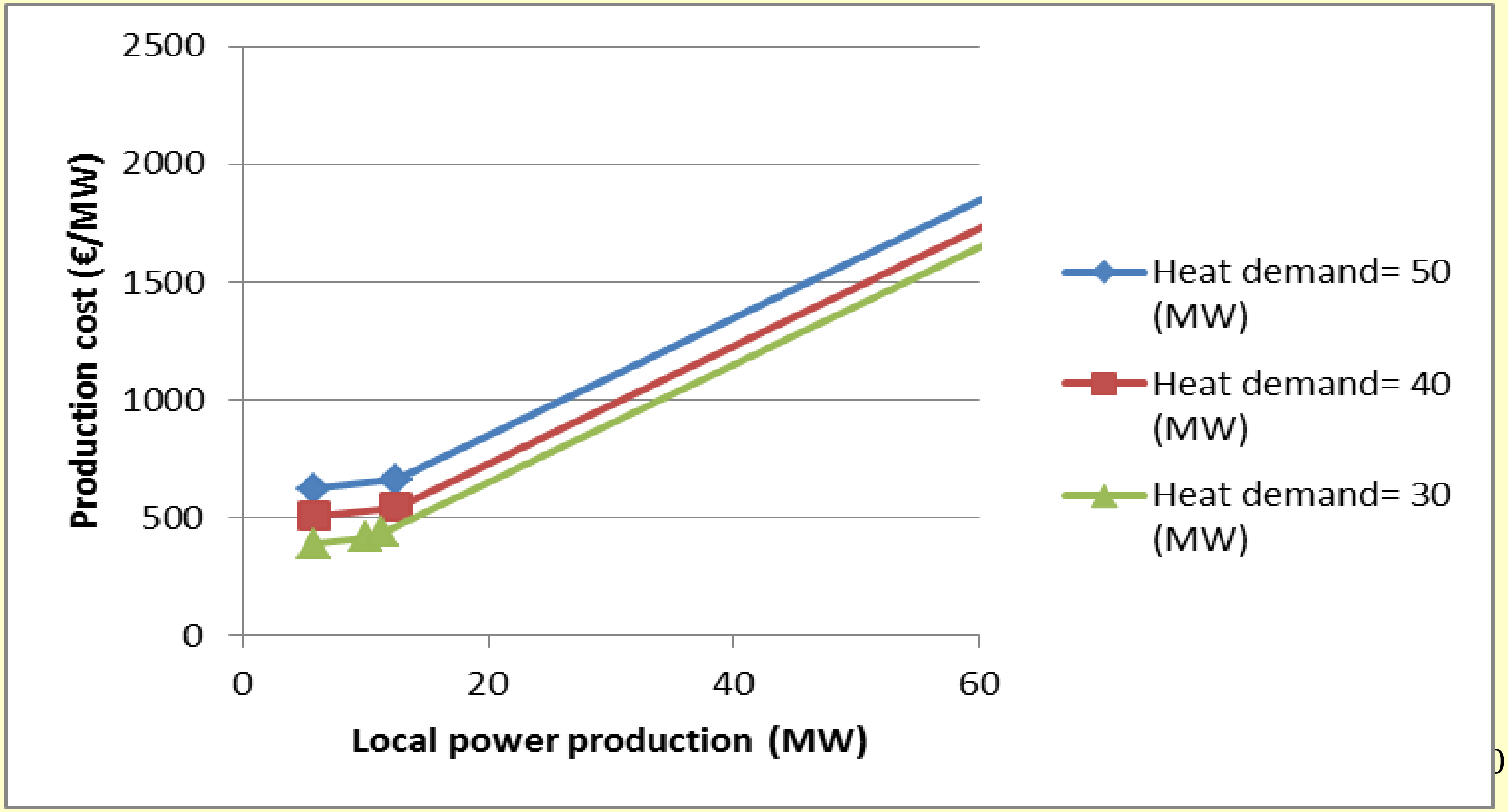
x1 <= 100; x2 <= 200; x3 <= 300;

x1, x2, x3 >= 0;

  - Note! One constraint is redundant, model equal with LP

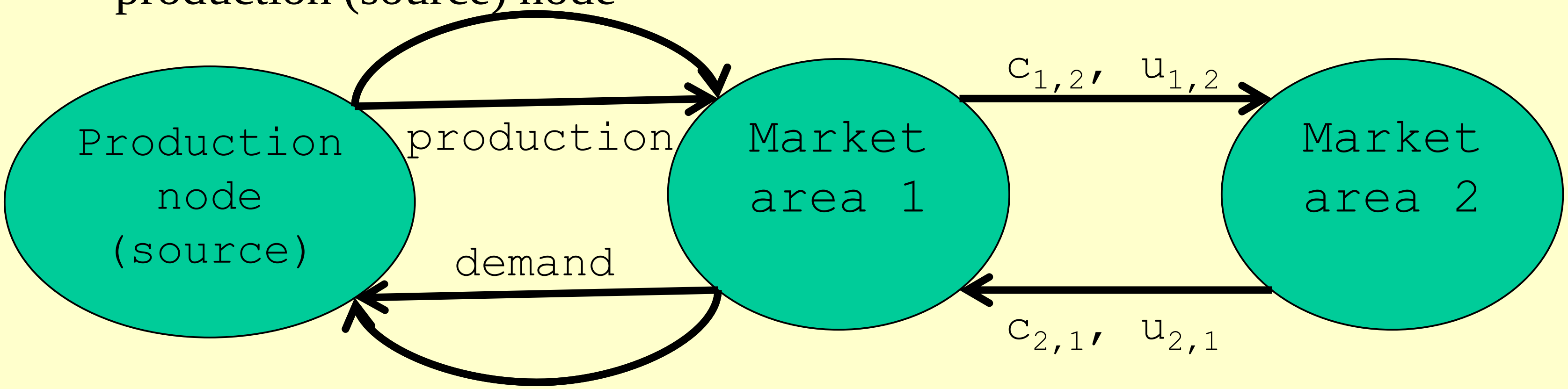
Combined power production and transmission problem

- Represent optimal production cost in each area as piecewise linear convex function of power production
  - Cost functions can be computed using parametric analysis on LP model for production in area (heat demand should be known)



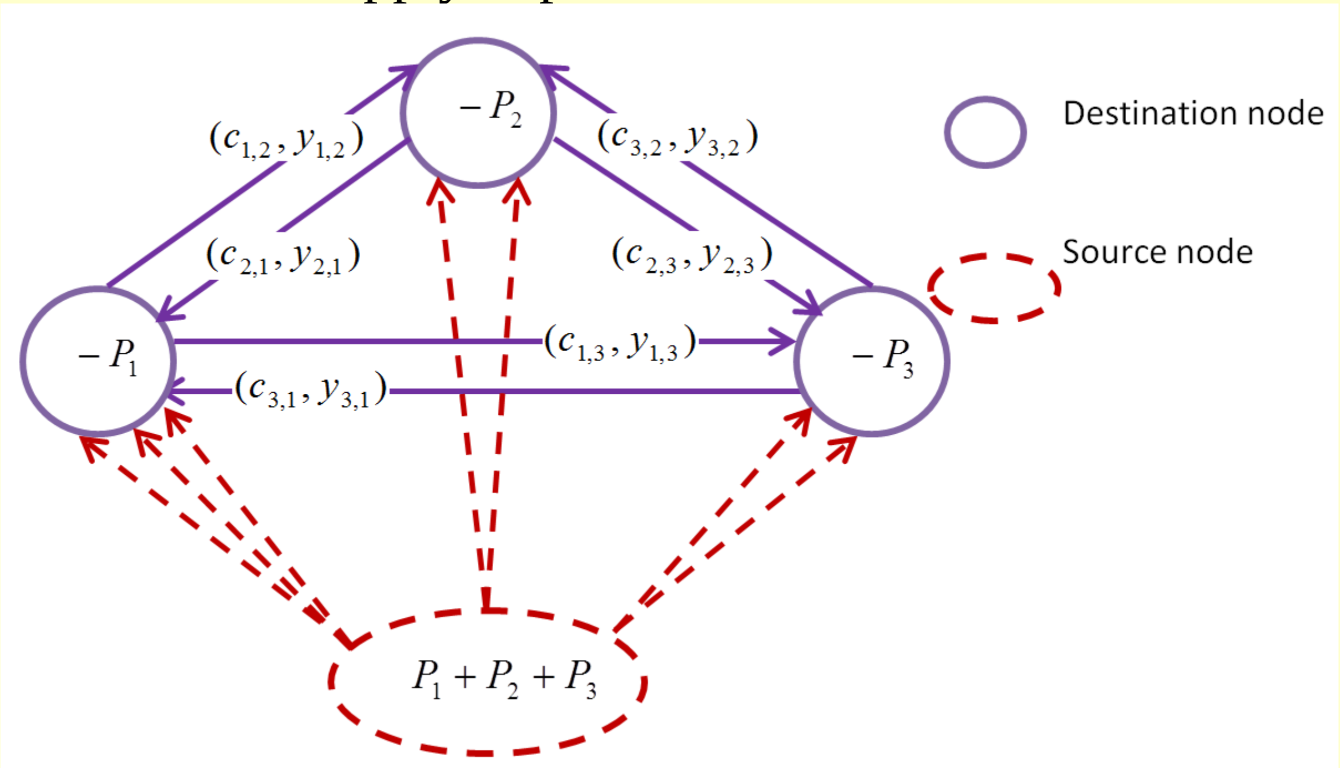
Transshipment model of power transmission across market areas

- A node is created for each market area *i* with fixed demand P_i
- Transmission lines are represented by directed arcs
  - Transmission capacity is represented by capacity limit u_{ij} for arc
  - Transmission loss is represesnted as a unit cost c_{ij} for arc
- Production capacity and elastic demand in each area
  - Production cost and demand are piecewise linear functions
  - Represented by multiple incoming/outgoing capacited arcs from a common production (source) node



Combined power production and transmission problem

- Transmission lines = (blue) transmission arcs with capacities&costs between areas
- Piecewise linear production costs = red production arcs from production node to corresponding area
- Combined fixed demand = supply at production node



Review questions

- Please review lecture material at home before next lecture and be prepared to answer review questions at beginning of next lecture.
  1. Is it necessary that the static models in dynamic programming are convex?
  2. Why should there not be negative costs/distances in the shortest path network?
  3. Why must the sum of all supplies/demands in a transshipment network equal zero?
  4. What kind of problems can be represented as a transshipment network flow model?
  5. What is the advantage of representing a problem as a transshipment network flow model?