NWE Stakeholder Forum in Brussels, 26 September 2012

NWE Day-Ahead Price Coupling

Algorithm

Agenda

1. Algorithm Development Milestones

2. PCR Algorithm Requirements Overview

3. PCR Algorithm Network Requirements
   - ATC line and Cumulative ATC
   - Flow Ramping and Cumulative Flow Ramping
   - Losses
   - Tariffs
   - Minimum Stable Flow
   - Net Position Ramping
   - Flow calculation
   - Flow-Based and Hybrid ATC/FB

4. PCR Algorithm PXs Requirements
   - Welfare Maximization
   - Prices
   - Hourly Orders
   - Block Orders
   - Complex Orders and PUN

Algorithm Development Milestones

Starting Point Selection – ended March 2011

- A decision was made in Sept 2010 to use one of these algorithms as a basis for the development of the PCR Coupling Algorithm;
- Each algorithm was assessed in regards with the currently implemented requirements and the ability to evolve as the PCR Coupling Algorithm;
- The CWE Coupling Algorithm was selected in March 2011;

Preparation of Industrialization – July 2011 – May 2012

- R&D work focusing on challenging requirements;
- Requirements and Specifications (regular exchanges with NWE TSOs and NRAs);
- Organization of industrialization phase (included planning, testing);

Algorithm Industrialization – June 2012 – December 2012

- Implementation of new requirements and enhancement of code maintainability;
- Sequential and incremental process of release deliveries;
- Final validation of production release December 2012;

PCR Algorithm Requirements Overview

Implementation Status

(1) Already implemented - The requirement is already in the R&D Algorithm Prototype which is industrialized and/or the PMB design and will be part of the first NWE release. The requirement has been budgeted.

(2) Implemented in first release compatible with NWE launch date – The requirement is not implemented in the R&D Algorithm Prototype, but will be part of the first NWE release. The requirement has been budgeted .

(3) Implemented in a next release after the Go live - The requirement will not be implemented for the launch of NWE. The requirement might be subject to R&D work (if mentioned in remarks) before it can be implemented in the Algorithm and/or PMB. The requirement has not been budgeted yet. The planning for implementation is not determined.

Capacity Allocation Requirements	Implementation Status
ATC Line	1
Cumulative ATC	3 / 1 (virtual bidding area)
Flow Ramping	1
Cumulative Flow Ramping	2
Long Term Nominations	3
Losses	1
Tariffs	1
Minimum Stable Flow	3
Net Position Ramping	1
Flow Calculation	2
Flow-Based and Hybrid ATC/FB	1
Welfare maximization	1

PXs Requirements	Implementation Status
Hourly Orders	1
Block Orders	1
Complex Orders and PUN	2
Branch-and-Bound	1
Java, Oracle	1
Optimization process	1

PCR Algorithm Requirements Overview

Changes for Market Participants

CWE Traders:
no change due to the algorithm (but for instance possible local introduction of new block order types already modeled in the algorithm)
CWE TSOs: no change due to the algorithm (but local projects such as Flow-Based remain)

Nordic Traders:
small change in acceptance condition of linked block orders based on if combined acceptance of several blocks leads to net welfare gain, rather than required for each block independently + small modifications of market rules (e.g. price indeterminacy)

Nordic TSOs:
cumulative flow ramping does not need virtual bidding areas any more + small modifications of market rules

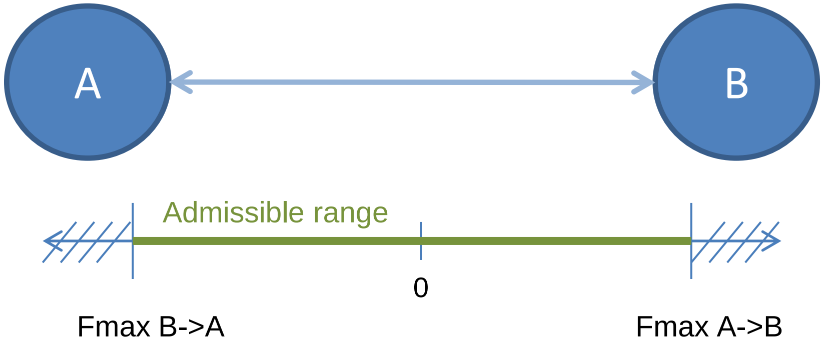
Capacity Allocation Requirements	Implementation Status
ATC Line	already in Sesam & Cosmos
Cumulative ATC	3 / 1 (virtual bidding area)
Flow Ramping	already in Sesam & Cosmos
Cumulative Flow Ramping	2
Long Term Nominations	3
Losses	already in Sesam & Cosmos
Tariffs	already in Sesam & Cosmos
Minimum Stable Flow	3
Net Position Ramping	already in Sesam & Cosmos
Flow Calculation	2 – already in Sesam
Flow-Based and Hybrid ATC/FB	already in Cosmos
Welfare maximization	already in Sesam & Cosmos

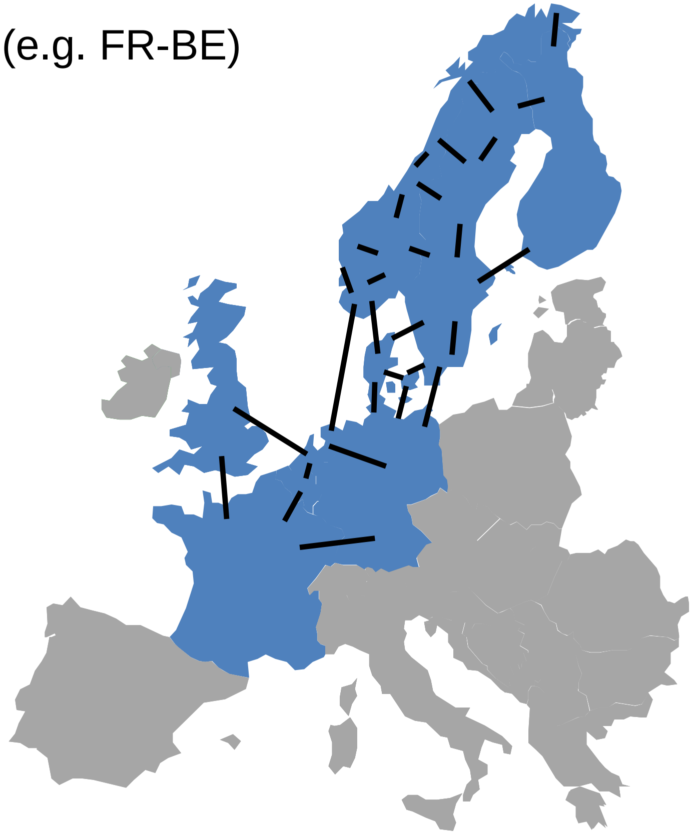
PXs Requirements	Implementation Status
Hourly Orders	already in Sesam & Cosmos
Block Orders	already in Sesam & Cosmos
Complex Orders and PUN	2
Branch-and-Bound	already in Sesam & Cosmos
Java, Oracle	already in Cosmos
Optimization process	already in Sesam & Cosmos

PCR Algorithm Network Requirements

ATC Network – Line Capacities

- Flows are bounded by ATCs i.e. limit capacities
- Lines can be real interconnectors (e.g. FR-UK)
- Lines can be a modeling with no direct physical reality (e.g. FR-BE)

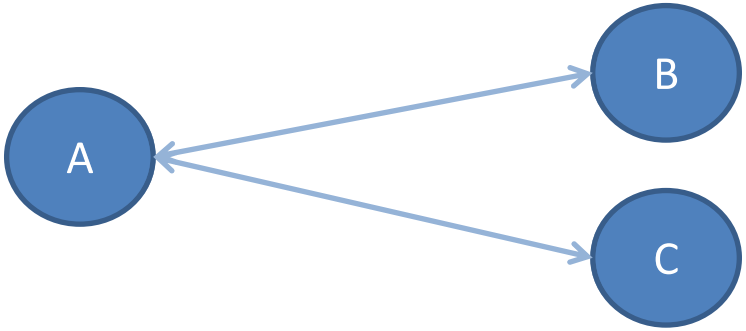


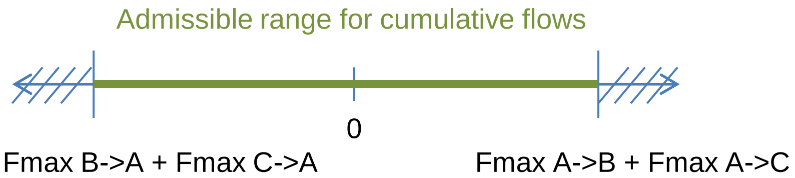


PCR Algorithm Network Requirements

ATC Network – Cumulative ATC

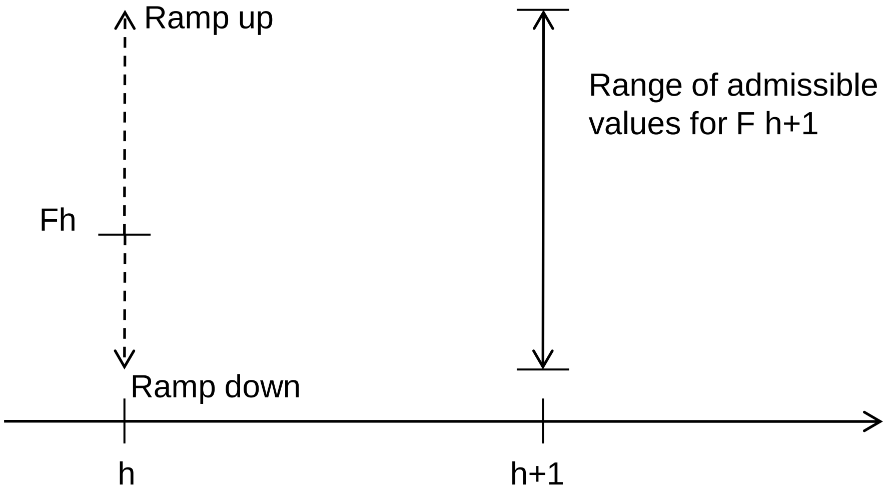
- Flows are limited on a set of lines





ATC Network – Flow Ramping

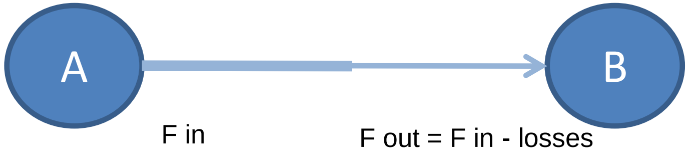
- Variation of flows are bounded by ramping limits
- Ramping constraints might trigger non optimal and adverse flows



PCR Algorithm Network Requirements

ATC Network – Losses

- Linear losses are allowed
- Losses have impact on volume and on prices



ATC Network – Tariffs

- Linear tariffs are allowed
- A line can have losses without tariff and conversely
- Tariff corresponds to an imposed price difference before starting to trade
- Tariff corresponds to the cost of using the cable

Net Position Ramping

- The variation of the net position (difference of supply and demand) of a bidding area can be limited; up and down, depending on constraints on network security
- Net postion ramping can be hourly (variation from one hour to next hour) or over the whole day

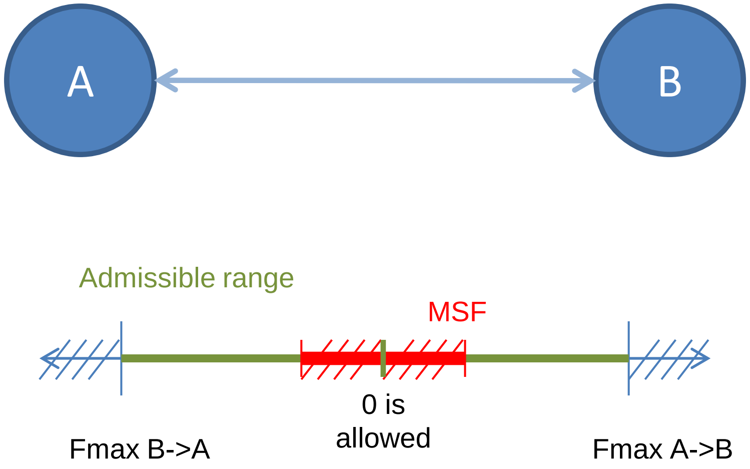
ATC Network – Flow Calculation

- Flows are output by the algorithm and can be used
- A post-processing solves flow indeterminacies

PCR Algorithm Network Requirements

ATC Network – Minimum Stable Flow

- Minimum Stable Flow corresponds to the Minimum Energy through the cable
- Can be used when the Algorithm will consider long term nominations (and not only day-ahead nominations)
- MSF can trigger adverse flows



ATC Network – Adverse Flow Summary

Adverse flows on ATC lines are only tolerated in cases where:

- The ramping constraint forces flow in a direction (forcing flow results in more welfare than sticking to non-adverse flows)
- The line capacity (ATC limit) is negative and this constraint is binding
- The line operates at precisely the minimum stable load flow
- Or the ramping constraint on net position is binding

PCR Algorithm Network Requirements

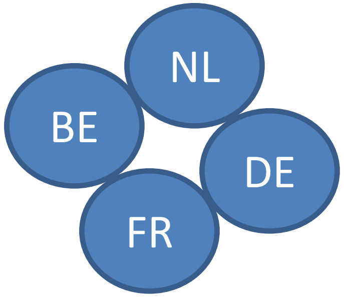
Flow-Based Network

- The modeling of the network can be more accurate, leaving more room to the market to determine the optimal configuration of exchanges

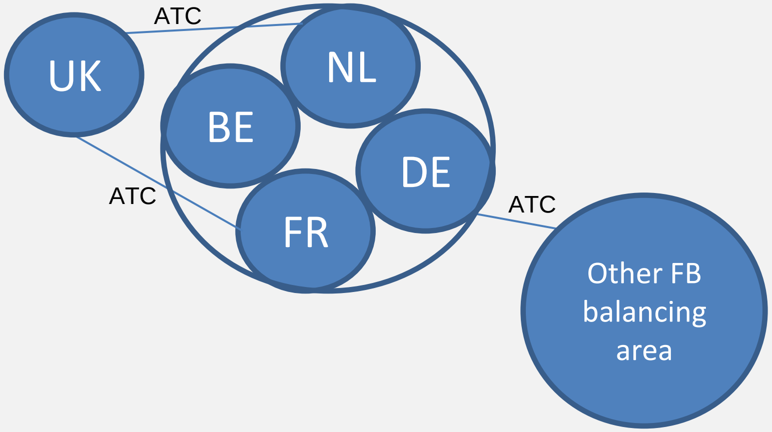
- In regards with the algorithm, flows are no longer calculated (flow-based substitutes to ATC network)

- Non intuitive effects can occur; forcing intuitive solutions remains possible

(e.g. NL exporting with higher price than BE and DE is a non-intuitive situation, which is possible under regular flow-based)



No flow in FB balancing area (ATC lines no longer existing)



Flow-Based Network – Hybrid Coupling ATC/FB

- Flow-based balancing area can be coupled with ATC networks of other bidding areas

- Standard hybrid coupling is supported by the algorithm

- Advanced hybrid coupling can be supported by means of virtual bidding areas

(purpose of advanced hybrid coupling is to avoid a priori ATC restrictions for lines connected with FB, although profitable for the market)

PCR Algorithm PXs Requirements

Welfare Maximization

- The model in the algorithm is formulated as a welfare maximization problem (Mixed Integer Quadratic Problem)
- Welfare is the sum of congestion rent and producer and consumer surplus

Block Orders

- span several consecutive hours
- are kill-or-fill
- in-the-money can be accepted or paradoxically rejected / block orders out-of-the-money must be rejected
- can be linked; flexible; profile block (different quantity for each hour)

Small change in acceptance condition of linked block orders based on if combined acceptance of several blocks leads to net welfare gain, rather than required for each block independently

Prices

- Negative prices are allowed
- Non harmonized price boundaries are possible
- Prices are always unique (the algorithm manages price indeterminacies)

Hourly Orders

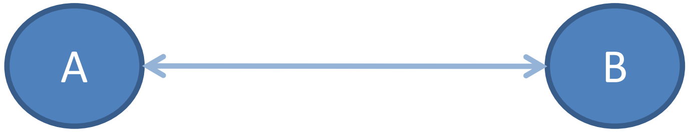
- Hourly orders in-the-money must be accepted / orders out-of-the-money must be rejected / orders at the money can be partially accepted
- Hourly curves can be stepwise, linear interpolated or hybrid

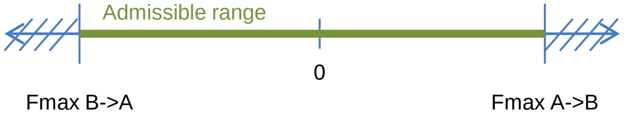
Complex Orders and PUN

Back-up

PCR Algorithm Network Requirements

ATC Network – Line Capacities





Additional features

- ATCs are sent by TSOs to coupling system
- ATCs are defined on an hourly basis
- Negative ATCs are possible, forcing flow in a given direction
- Flows are intuitive and always go from the cheaper bidding area to the more expensive biddning area (if no ramping, no negative ATC, no minimum stable flow)

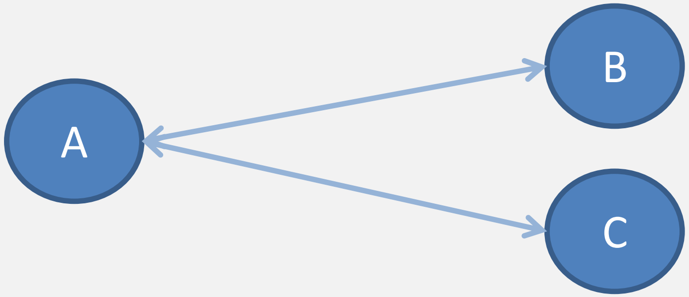
Energy Balance (simple ATC line without tariff and losses)

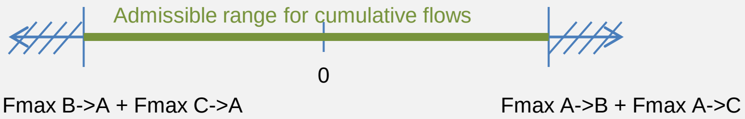
Every bidding area is balanced in energy
The sum of net positions of bidding areas is zero

Price Properties (simple ATC line without tariff and losses)

If the line is not congested, price A = price B
If price A ≠ price B, the line is congested (i.e. meets one bound)

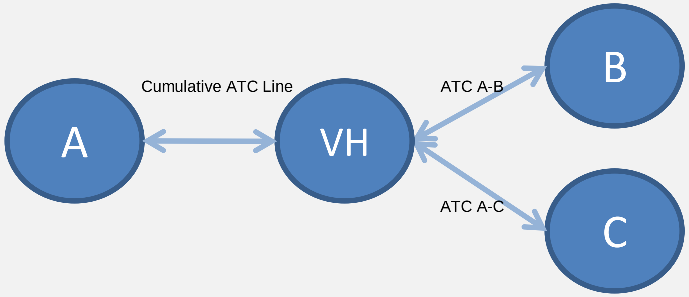
ATC Network – Cumulative ATC





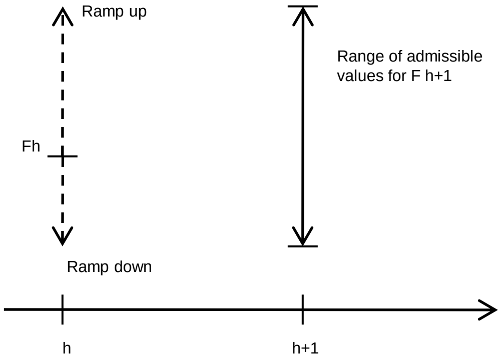
Additional features

- Cumulative ATCs are sent by TSOs to coupling system
- Cumulative ATCs are defined on an hourly basis
- Cumulative ATCs will not be implemented by PCR Algorithm launch
- Current implementation uses virtual hub in between



PCR Algorithm Network Requirements

ATC Network – Flow Ramping

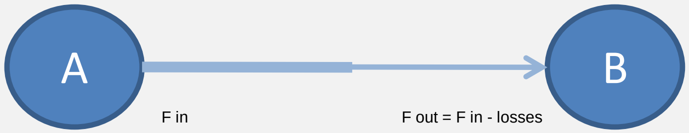


Additional features
- Ramping limits are sent by TSOs to coupling system
- Ramping limits are defined on an daily basis (current CWE implementation) or on an hourly basis (PCR implementation)
- Ramping up and down can be different
- **Cumulative Ramping** will be available by PCR launch

General (cumulative) ramping modelling
Notations
d – « down »
u – « up »
h – « hour »
F – flow
RL – ramping limit

$$-\mathrm{RL}^{\mathrm{Lines}}_{\mathrm{d,h}} \le \sum_{i=1}^{n}\left[\left(\mathrm{F}^{\mathrm{Line}_i}_{\mathrm{u,h}} - \mathrm{F}^{\mathrm{Line}_i}_{\mathrm{d,h}}\right) - \left(\mathrm{F}^{\mathrm{Line}_i}_{\mathrm{u,h-1}} - \mathrm{F}^{\mathrm{Line}_i}_{\mathrm{d,h-1}}\right)\right] \le \mathrm{RL}^{\mathrm{Lines}}_{\mathrm{u,h}}$$

ATC Network – Losses



Additional features
- Losses are applied on the exporting market (which is the cheaper market, which makes sense under welfare optimization)
- Losses percentage is sent by TSOs as a parameter of the line

Energy Balance
The sum of net positions plus losses is equal to zero

Price Properties
Notations
mcp – market coupling price
to/from – refers to the bidding importing/exporting bidding area
loss – loss factor (between 0 and 1)

$\mathrm{mcp}_{\mathrm{to}}$ (1-loss) - $\mathrm{mcp}_{\mathrm{from}}$ = 0 when no congestion
$\mathrm{mcp}_{\mathrm{to}}$ (1-loss) - $\mathrm{mcp}_{\mathrm{from}}$ > 0 when line is congested

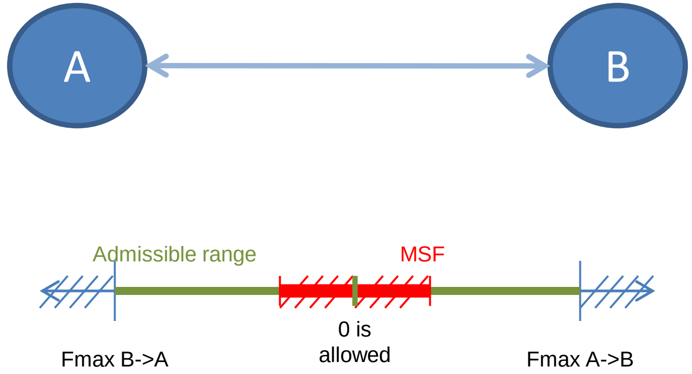
ATC Network – Tariffs

Additional features
Tariff is sent by TSOs as a parameter of the line

Price Properties
$\mathrm{mcp}_{\mathrm{to}}$ - $\mathrm{mcp}_{\mathrm{from}}$ = flow_tariff when no congestion
$\mathrm{mcp}_{\mathrm{to}}$ - $\mathrm{mcp}_{\mathrm{from}}$ > flow_tariff when line is congested

PCR Algorithm Network Requirements

ATC Network – Minimum Stable Flow



Price Properties
Notations
l – refers to a given line
other notation: see previous slides

In case the capacity constraint is non-binding in both directions (i.e. uncongested), we want the relevant price constraint to be tight too:

$F_{u,l,h}$> $MSF_{u,l,h}$ implies $(1\text{-}loss_{u,l,h})mcp_{to} - mcp_{from}$ = flow $tariff_{u,l,h}$

$F_{d,l,h}$> $MSF_{d,l,h}$ implies $(1\text{-}loss_{d,l,h})mcp_{from} - mcp_{to}$ = flow $tariff_{d,l,h}$

In case the capacity constraint is binding in one direction, the equality becomes an inequality (e.g. $(1\text{-}loss_{u,l,h})mcp_{to} - mcp_{from}$ > flow $tariff_{u,l,h}$)

Net Position Ramping

Additional features
•Net position ramping limits are sent by TSOs
•Net position can trigger adverse flows

Flow-Based Network

Additional features
Notations
PTDF – power transfer distribution factor
z – refers to the bidding area
e – net position
α, k – refers to the constraint
maxF –bound of the constraint
cp – constraint shadow price

PTDF matrices and bounds are sent by TSOs
Shadow prices for every constraint are output by the algorithm

$$\sum_{z\in Z} PTDF_{\alpha}^{z} \cdot e_z \leq maxF_{\alpha}$$

Energy Balance – Flow-Based Constraint
Net positions are linked and constrained by PTDFs

Price Properties
Price difference under regular flow-based is proportional to PTDF difference and shadow prices

$$mcp_i - mcp_j = \sum_{k\in K} \left(PTDF_{k,j,h} - PTDF_{k,i,h}\right) cp_{k,h}$$

In case intuitivity is enforced, price differences follow regular ATC pricing rules

PCR Algorithm PXs Requirements

Particular Features

- Integrated Branch-and-Bound (managed by CPLEX)
- Finds the welfare-optimal solution
- JAVA implementation
- Oracle DB
- Precise indeterminacy rules for price & volume
- Strategy to limit the PRB loss

Branch-and-Bound

- By means of branch-and-bound and dedicated cutting planes, addition at nodes with PAB
- B&B managed by CPLEX, with control through callbacks on
  - Branching strategies
  - Node selection strategies
  - Cutting plane addition
- Parallelization possible for increased computing power (Parallel CPLEX)

Java Implementation

- Essentially used for setting up MIPs for CPLEX
  - No heavy algorithm implemented in JAVA
  - Heavy algorithmic code is within CPLEX that is an excellent implementation in C/C++
- Java enables very fast and very robust prototyping and implementation
- Easy to integrate with other modules (written in different languages)

Oracle Database

- The algorithm interacts with outside world by reading from and writing to Oracle DB tables
  - Modular architecture
  - Professional technologies
- Other DB technologies possible as far as the algorithm is concerned

PCR Algorithm PXs Requirements

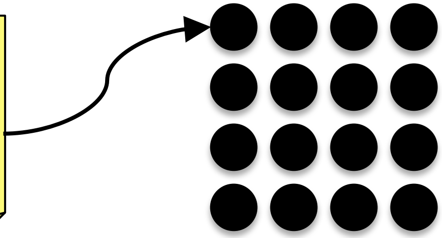
Optimization process

- Branch-and-Bound method is a way to
  - Search among all these block selections in a structured way
  - Find feasible solutions quickly
  - Prove early that large groups of these selections cannot hold good solutions

- The idea is as follows
  - Try first without the kill-or-fill requirement
  - If the solution happens to have no fractional block, OK
  - If it has, then
    - Select one block which is fractional
    - Create two subproblems (called branches)
      - One where the block is killed
      - One where the block is rejected
    - Continue to explore until there is no unexplored branch

A (naïve) algorithm

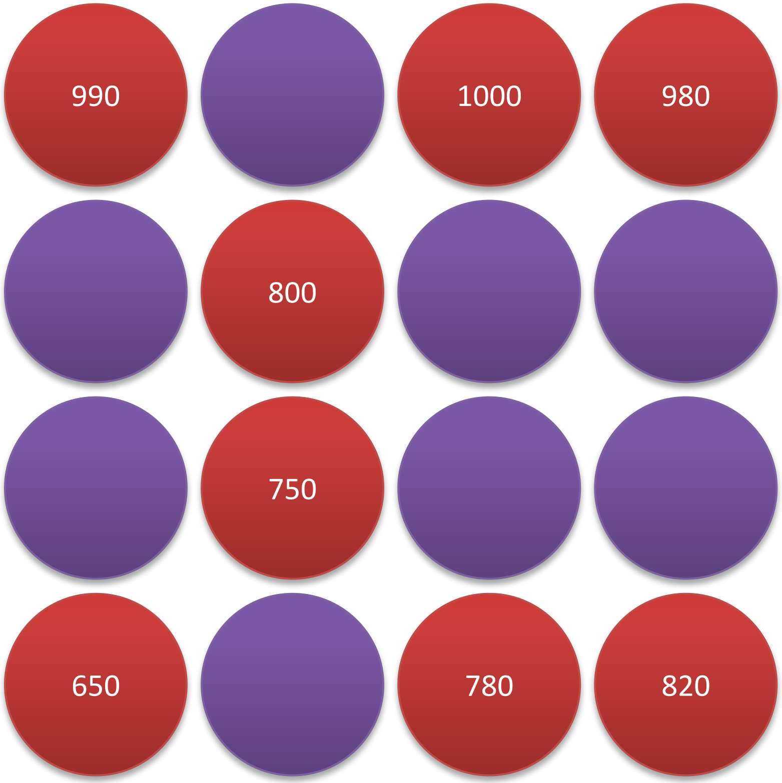
Each ball corresponds to a specific block selection. Assume these balls exhaustively list all possible block selections (i.e. there are 4 blocks, hence 2^4 = 16 block selections)



For each block selection

- Check whether it creates PABs
- Among the selections without PABs, return the one giving the largest welfare
- But this is not efficient: if there are 100 blocks, there are 2^100 ≈ 1030 possibilities

- Bounding
  - Suppose we already have a good valid solution
  - When exploring one branch, we might see that even if we have not yet reached an integer solution, the current solution is not better than the best one we already have
  - Then we can decide to stop exploring this branch, since it will never lead to a better welfare than the current best solution


990
1000
980
800
750
650
780
820