ALL NEMO COMMITTEE

EUPHEMIA Public Description

Single Price Coupling Algorithm

18th December 2025

Contents

1. Introduction

The Algorithm methodology in force, adopted by ACER Decision 04/2020, states the following in Art. 4.18:

"All NEMOs shall create and maintain a document with the detailed description of the price coupling algorithm, including the description of calculation of scheduled exchanges in accordance with the methodology for calculating scheduled exchanges for the day-ahead timeframe. This document shall be published and kept updated with every new version of the price coupling algorithm. The document shall be publicly available by all NEMOs on a public webpage."

The main purpose of this document is to seek legal compliance with the abovementioned mandate. Furthermore, this public description aims at disseminating and facilitating the understanding of the single price coupling algorithm among stakeholders and the wider public.

Additionally, the MCO Plan approved by all EU National Regulatory Authorities on 26 June 2017 confirms the adoption of the "Price Coupling of Regions" (PCR) solution as the basis for the single day-ahead coupling.

This solution is used as the solution to both the SDAC market, as well as the SIDC IDAs.

Price Coupling of Regions (PCR) project is an initiative of eight Power Exchanges (PXs): EPEX SPOT, GME, HEnEx, Nord Pool EMCO, OMIE, OPCOM, OTE and TGE covering the electricity markets in Austria, Belgium, Bulgaria, Czech Republic, Croatia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Italy, Ireland, Latvia, Lithuania, Luxembourg, the Netherlands, Northern Ireland, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden . PCR is implemented in SDAC, following the merge of the MRC region as well as the 4M Market Coupling (4M MC).

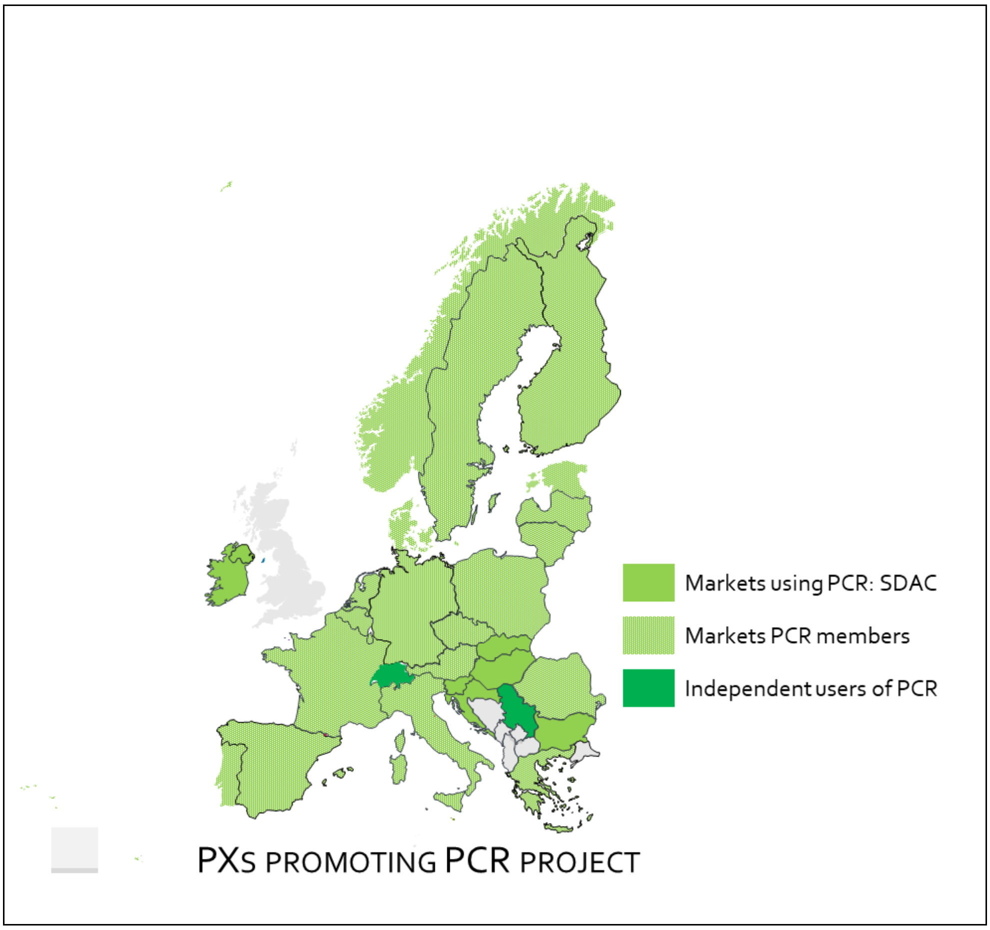


Figure 1 – PXs promoting PCR project

One of the key achievements of the PCR project is the development of a single price coupling algorithm, commonly known as EUPHEMIA (acronym for Pan-European Hybrid Electricity Market Integration Algorithm). Since February 2014, EUPHEMIA has been progressively used to calculate energy allocation and electricity prices across Europe, maximizing the overall economic surplus and increasing the transparency of the computation of prices and flows.

In the past, several algorithms were used locally by the involved PXs. All these algorithms (COSMOS, SESAM, SIOM and UPPO) have been focusing on the products and features of the corresponding PX, but none was able to cover the whole set of requirements. This made the implementation of the new algorithm (EUPHEMIA) necessary, to cover all the requirements at the same time and give solutions within a reasonable time frame.

Total market economic surplus is the sum across all bidding zones of consumer surplus, producer surplus, and congestion income. Breaking this down further, each component can be expressed more explicitly.

Consumer Surplus = Sum over all demand orders of [(Maximum price the buyer is willing to pay minus the clearing price) multiplied by the volume accepted]

Producer Surplus = Sum over all supply orders of [(Clearing price minus minimum price the seller is willing to accept) multiplied by the volume sold]

Congestion Income = Sum of the product of the scheduled exchanges across each border with the price difference between the markets. Note this equates to the sum of Clearing price multiplied by [Total demand volume minus total supply volume in the zone] over all bidding zones.

Considering a basic problem with stepwise curves only, no tariffs, and one period of 60 minutes, it can be mathematically shown as follow:

$$\sum_{bz\in BZ}\left[\underbrace{\sum_{b\in B_{bz}}(p_b-\pi_{bz})v_b}_{CS}+\underbrace{\sum_{s\in S_{bz}}(\pi_{bz}-p_s)v_s}_{PS}+\underbrace{\pi_{bz}\left(\sum_{b\in B_{bz}}v_b-\sum_{s\in S_{bz}}v_s\right)}_{CI}\right]$$

$$=\sum_{bz\in BZ}\left[\sum_{b\in B_{bz}}p_b v_b-\sum_{b\in B_{bz}}\pi_{bz}v_b+\sum_{s\in S_{bz}}\pi_{bz}v_s-\sum_{s\in S_{bz}}p_s v_s+\sum_{b\in B_{bz}}\pi_{bz}v_b-\sum_{s\in S_{bz}}\pi_{bz}v_s\right]$$

$$=\sum_{bz\in BZ}\left[\underbrace{\sum_{b\in B_{bz}}p_b v_b}_{Consumer\ benefit}-\underbrace{\sum_{s\in S_{bz}}p_s v_s}_{Producer\ costs}\right]$$

with

- *BZ* the set of bidding zones,
- B_{bz} the set of demand orders of bidding zone *bz*,
- p_b the maximum price that order *b* is keen to pay,
- v_b the volume accepted for order *b*,
- S_{bz} the set of supply orders of bidding zone *bz*,
- p_s the minimum price that order *s* is keen to receive,
- v_s the volume accepted for order *s*,
- π_{bz} the price cleared for bidding zone *bz.* It becomes apparent when examining the expanded equation, which includes four terms involving the clearing price:

Negative: "Clearing price times demand volume" (from consumer surplus)

Positive: "Clearing price times supply volume" (from producer surplus)

Positive: "Clearing price times demand volume" (from congestion income)

Negative: "Clearing price times supply volume" (from congestion income)

When grouped together, the **negative terms cancel out the positive ones**, resulting in a simplified expression. Thus, the **congestion income terms** are **removed algebraically**. This is why "**Annex C: Mathematical Approach**" does not contain any terms related to congestion rent.

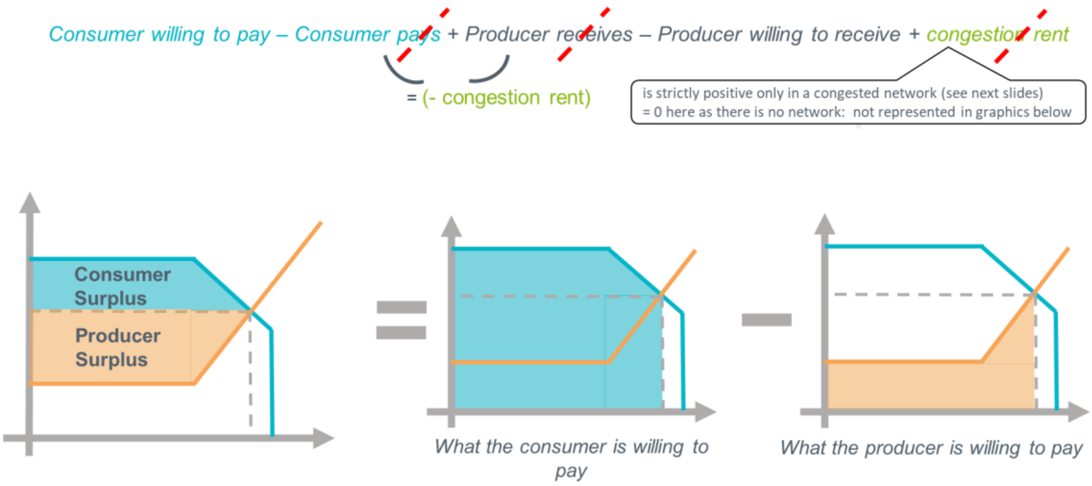
This final form demonstrates that **total economic surplus is the sum of consumer benefit minus producer costs across all bidding zones**. This mathematical transformation unveils a significant economic insight into how the model generates value. The intricate three-part economic surplus function, which separately accounts for consumer surplus, producer surplus, and congestion rent, ultimately reduces to a single measure: the difference between society's valuation of electricity and its production cost. This convergence happens because the market clearing process inherently balances the congestion rent terms across the system.

Congestion rent is implicitly considered in Euphemia in the area of supply and demand (Image 1). It is not explicitly added to the economic surplus objective itself. Instead, it naturally arises from the solution when Market Clearing Prices between interconnected bidding zones differ due to cross-zonal transmission capacity constraints that are binding. Congestion rent is the revenue generated from this price spread multiplied by the inter-zonal flow over the congested interconnectors.

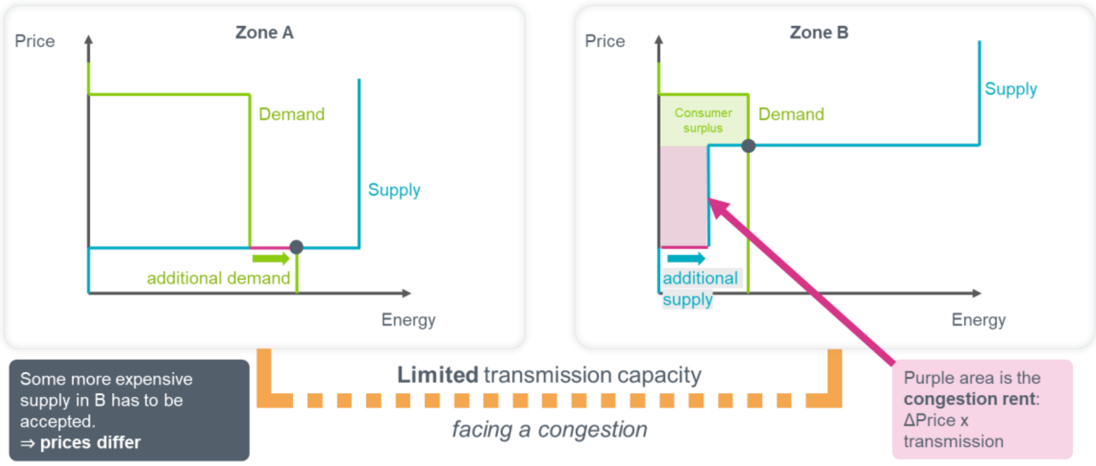
Mathematically, the dual variables associated with these cross-zonal flow constraints yield the shadow prices of congestion. In the absence of network constraints, the congestion rent is zero.

Graphically, this can be represented as follows:

Economic Surplus = Consumer Surplus + Producer Surplus + Congestion rent



In the figure below, a congested line is modelled as an additional supply order for the import zone and an additional demand order for the export zone. Both orders are illustrated in purple.



This figure illustrates two bidding areas connected by a transmission line with limited capacity. Area A has excess supply, while Area B requires imports.

In an uncongested transmission system, electricity flows freely from Area A (lower-cost generation) to Area B (higher demand). This arbitrage process results in a single market-clearing price across both areas. However, when the transmission line reaches its physical capacity limit, it becomes a binding constraint that prevents additional low-cost electricity from Area A from reaching Area B.

The transmission constraint creates separate market-clearing prices in each area:

In Area A (the export zone), the power flow to Area B acts as additional demand, shown in purple. The market clears at the intersection of local supply and total demand (local demand plus export to Area B).

In Area B (the import zone), the power flow from Area A acts as additional supply, shown in purple. The market clears at the intersection of total supply (local supply plus import from Area A) and local demand.

This results in a price differential where Area B's price exceeds Area A's price due to the transmission constraint acting as an economic barrier. The purple shaded area represents congestion rent, calculated as the price difference multiplied by the constrained power flow. This captures the economic value of the scarce transmission capacity.

If we examine the first figure more closely, the flow going from Area A multiplied by (Price A - Price B) represents what consumers pay, while the flow coming to Area B multiplied by (Price B - Price A) represents what producers receive. These components compose the congestion rent but have the same amount with different signs, which cancel each other out.

To conclude, the key insight is that transmission constraints create relative scarcity in the import zone and relative abundance in the export zone, generating price spreads that reflect the marginal value of additional transmission capacity. This explains why congestion rent appears in economic surplus calculations—it represents real economic value created by efficiently allocating scarce transmission resources between areas with different supply-demand balances.

2. Day-Ahead Market Coupling Principle

Market Coupling (MC) is a way to join and integrate different energy markets into one coupled market. In a coupled market, demand and supply orders in one market are no longer confined to the local territorial scope. On the contrary, in a market coupling approach, energy transactions can involve sellers and buyers from different areas, only restricted by the electricity network constraints.

The main benefit of the Market Coupling approach resides in improving of the market liquidity combined with the beneficial side effect of less volatile electricity prices. Market coupling is beneficial for market players too. They no longer need to acquire transmission capacity rights to carry out cross-border exchanges, since these cross-border exchanges are given as the result of the MC mechanism. They only have to submit a single order in their market (via their corresponding PX) which will be matched with other competitive orders in the same market or other markets (provided the electricity network constraints are respected).

3. Introducing EUPHEMIA

EUPHEMIA is the algorithm that has been developed to solve the problem associated with the coupling of the day-ahead power markets in the PCR region.

First, Market participants start by submitting their orders to their respective power Exchange. All these orders are collected and submitted to EUPHEMIA that has to decide which orders are to be executed and which orders are to be rejected in concordance with the prices to be published such that:

- The *economic surplus* (consumer surplus + producer surplus + *congestion rent* across the regions) generated by the executed orders is maximal.
- The power flows induced by the executed orders, resulting in the *net positions* do not exceed the capacity of the relevant network elements.

EUPHEMIA handles standard and more sophisticated order types with all their requirements. It aims at rapidly finding a good first solution from which it continues trying to improve and increase the overall economic surplus. EUPHEMIA is a generic algorithm: there is no hard limit on the number of markets, orders or network constraints; all orders of the same type submitted by the participants are treated equally.

The development of EUPHEMIA started in July 2011 using one of the existing local algorithms COSMOS (being in use in CWE since November 2010) as starting point. The first stable version able to cover the whole PCR scope was internally delivered one year after (July 2012). Since then, the product has been evolving, including both corrective and evolutionary changes. On the 4th of February 2014, EUPHEMIA was used for the first time in production to couple the North Western Europe (NWE) in common synchronized mode with the South-Western Europe. One year later, on the 25th of February 2015, GME was successfully coupled. On the 21st of May 2015, the Central Western Europe was coupled for the first time using Flow-based model. On 20 November 2014 the 4M MC coupling was launched coupling the markets of Czech Republic, Hungary, Romania and Slovakia. The 4M MC coupling was merged with MRC on 17 June 2021. This was followed by the go-live of the Core Flow-Based Market Coupling project on 8 June 2022 and Day-Ahead Market Coupling on Croatian – Hungarian border on the same date. The 15-minute MTU in the Single Day-Ahead Coupling (SDAC) went live successfully on 30 September 2025, for delivery day 1 October 2025.

In the two following chapters, we explain which network models and market products can be handled by EUPHEMIA. Chapter 6 gives a high-level description of how EUPHEMIA works.

4. Power Transmission Network

EUPHEMIA receives information about the power transmission network which is enforced in the form of constraints to be respected by the final solution.

This information is provided by TSOs as an input to the algorithm.

4.1. Already Allocated Capacities

The usage of intraday market auctions (IDA) requires the introduction of already allocated capacity (AAC), in order to indicate the flows referring to a given line (or given lines, for example, in case of line set) for a specific flow date, for a specific period and to a session (or sessions) run previously than current one.

In example, AAC for line *LINE_ID,* period *p*, Delivery Day *D* IDA 1 can refer to flow assigned to line *LINE_ID*, period *p*, Delivery Day *D* obtained in the DA calculation.

AAC, defined in both Up and Down directions, is relevant in case of:

- Losses
- Ramping Limits assigned to:
  - Lines
  - Line Sets
  - Bidding Zones like negative losses

Let’s considered the following examples:

Example 1: AAC Disabled

- Single period market
- 2 Bidding Zones (A;B)
- Line direction: A→B

- Capacity Up: 10000 MW; Capacity Down: 10000 MW
- AAC Up:10 MWh; AAC Down: 0 MWh (note that in following example, AAC is Disabled, meaning it will be ignored by Euphemia)
- Loss Up:10%; Loss Down:10%
- OBK:
  - 10 MWh@30 €/MWh Demand in BZ A
  - 100 MWh@3 €/MWh Supply in BZ B

Being AAC ignored:

- FLOW_IN_UP: 0 MWh
- FLOW_IN_DOWN: 11.11 MWh
- FLOW_OUT_UP: 0 MWh
- FLOW_OUT_DOWN: 10 MWh
- IDA losses=1.11
- Total losses=2.11 MWh (1.11 for IDA only; 1 MWh from AAC)

Example 2: AAC Enabled

Now, given the same input set of Case 1, Let's enable the AAC.
In this case, before assigning flows in the opposite AAC direction, Euphemia will absorb existing AAC, resulting in:

- FLOW_IN_UP: -10 MWh
- FLOW_IN_DOWN: 0 MWh
- FLOW_OUT_UP: -9 MWh
- FLOW_OUT_DOWN: 0 MWh
- IDA losses=-1 MWh
- Total losses=0 MWh

from the example above, we can see that because of AAC it can be justified to allow negative flows that cancel existing AAC Up flows, minimizing losses on assigned line.

AAC and ramping limits: Lines

AACs becomes relevant also in ramping limits constraints, where they should be applied when calculating the delta flow between one period and previous one.
Given following input data:

- Two periods market
- 2 Bidding Zones(A;B)
- Line direction: A→B
- Period 1:
  - Capacity Up: 3 MW; Capacity Down: 10000 MW
  - AAC Up: 0 MWh; AAC Down: 0 MWh
  - No losses
  - Ramping limit up: 99999 MW; Ramping limit down: 99999 MW
- Period 2:
  - Capacity Up: 10000 MW; Capacity Down: 10000 MW
  - AAC Up: 7 MWh; AAC Down: 0 MWh
  - No losses
  - Ramping limit up: 11 MW; Ramping limit down: 99999 MW
- OBKs (identical for period 1 and period 2):

- 100 MWh@3 €/MWh Supply in BZ A
- 10 MWh@30 €/MWh Demand in BZ B

Results will be:

- Period 1, FLOW_IN_UP: 3 MWh
- Period 2, FLOW_IN_UP: 7 MWh
- Period 2 TOTAL FLOW - Period 1 TOTAL FLOW: ($FLOW\_IN\_UP_2$+AAC_UP_2)-($FLOW\_IN\_UP_1$+AAC_UP_1)=(7+7)-(3+0)=11 MWh➔Rampling Limit up not violated

Note that on the case on which AAC delta between one period and previous one are not complaint with ramping limits, in order to not invalidate the session due to infeasible input data, Euphemia will calculate and apply a *slack* variable in order to produce final results.
In example:

- Two periods market
- 2 Bidding Zones(A;B)
- Line direction: A→B
- Period 1:
  - Capacity Up: 3 MW; Capacity Down: 10000 MW
  - AAC Up: 0 MWh; AAC Down: 0 MWh
  - No losses
  - Ramping limit up: 99999 MW; Ramping limit down: 99999 MW
- Period 2:
  - Capacity Up: 10000 MW; Capacity Down: 10000 MW
  - AAC Up: 30 MWh; AAC Down: 0 MWh
  - No losses
  - Ramping limit up: 11 MW; Ramping limit down: 99999 MW
- OBKs (identical for period 1 and period 2):
  - 100 MWh@3 €/MWh Supply in BZ A
  - 10 MWh@30 €/MWh Demand in BZ B

In this case, starting point is not compatible with current ramping limit: Delta AAC = 30 MWh>Ramping limit (11 MWh). For this reason, Euphemia will assign to involved line a slack ramping up of 19 MWh to period 2, in order to contain AAC variation. Final results will be:

- Period 1, FLOW_IN_UP: 3 MWh
- Period 2, FLOW_IN_UP: 3 MWh
- Period 2 TOTAL FLOW - Period 1 TOTAL FLOW: ($FLOW\_IN\_UP_2$+AAC_UP_2-SLACK RAMPING UP_2)-($FLOW\_IN\_UP_1$+AAC_UP_1)=(3+30-19)-(3+0)=11 MWh➔Rampling Limit up not violated

AAC and ramping limits: Line set

In this example for illustrative purposes, we assume the total AAC of the line set to be inferior to the sum of the AACs of the different lines making up the line set.

Same principle of which AAC applied to single lines is applied to ramping limit validation for line set.
Note that for ramping limit applied to a line set, only AAC associated to lineset itself will be considered, ignoring the one(s) associated to lines which compose it.

In example:

- Two periods market
- 3 Bidding Zones(A;B;C)
- 3 Lines: A→B; A→C; B→C
- For each line, and period:
  - Infinite capacity in both directions
  - Infinite ramping limits in both directions
- For line A→B:
  - AAC $DOWN_1$: 4 MWh
  - AAC $DOWN_2$: 10 MWh
- A→B and A→C are assigned to line set 1:
  - Period 1:
    - Capacity Up: 4 MW
    - Capacity Down: 4 MW
    - Ramping Limit Up: 99999 MW
    - Ramping Limit Down: 99999 MW
    - AAC Up: 0 MWh
    - AAC Down: 3 MWh
  - Period 2:
    - Capacity Up: 99999 MW
    - Capacity Down: 99999 MW
    - Ramping Limit Up: 7 MW
    - Ramping Limit Down: 7 MW
    - AAC Up: 0 MWh
    - AAC Down: 5 MWh
- OBKs (identical for period 1 and period 2):
  - 10 MWh@30 €/MWh Demand in BZ A
  - 100 MWh@3 €/MWh Supply in BZ B
  - 100 MWh@3 €/MWh Supply in BZ B

Results will be:
- Period 1:
  - Line A→B FLOW_IN_DOWN: 2 MWh
  - Line A→C FLOW_IN_DOWN: 2 MWh
  - Line B→C FLOW_IN_DOWN(UP): 0 MWh
- Period 2:
  - Line A→B FLOW_IN_DOWN: 4.5 MWh
  - Line A→C FLOW_IN_DOWN: 4.5 MWh
  - Line B→C FLOW_IN_DOWN(UP): 0 MWh
- Period 2 LINE SET FLOW - Period 1 LINE SET FLOW: $\sum FLOW\_IN\_DOWN_2+LINE\_SET\_AAC\_DOWN_2-\sum FLOW\_IN\_DOWN_1+LINE\_SET\_AAC\_DOWN_1)=4.5+4-5+5-2-2-3=7$ MWh➔Line set Rampling Limit Down not violated

AAC and ramping limits: Bidding Zones

On the case on which a bidding zone is subjected to periodic (or daily) ramping limits, AAC should be considered in determining net position change between a period and previous one.
In example:

- Two periods market
- 3 Bidding Zones (A;B;C)
- 3 Lines: A→B; A→C; B→C
- Period 1:
  - A→B Capacity Down: 4 MW
  - A→C Capacity Down: 4 MW
- For other lines, and periods:
  - Infinite capacity in both directions
  - Infinite ramping limits in both directions
- For line A→B:
  - AAC $DOWN_1$: 4 MWh
  - AAC $DOWN_2$: 10 MWh
- Bidding zone A:
  - Ramping Limits Up period 1: infinite
  - Ramping Limits Down period 1: infinite
  - Ramping Limits Up period 2: 7 MW
  - Ramping Limits Down period 2: 7 MW
- OBKs (identical for period 1 and period 2):
  - 100 MWh@30 €/MWh Demand in BZ A
  - 100 MWh@3 €/MWh Supply in BZ B
  - 100 MWh@3 €/MWh Supply in BZ B

Results will be:

- Period 1, BZ A: -8 MW
- Period 2, BZ A: -9 MW
- Net position delta: -9-10-(-8-4)=-7 MWh→Ramping constraint not violated

4.2. Bidding Zones

A *bidding zone* (previously called *bidding area*, but the two are synonyms) corresponds to a geographical area to which network constraints are applied. Consequently all submitted orders in the same bidding zone will necessarily be subjected to the same unique clearing price. EUPHEMIA computes a market clearing price for each *bidding zone* and each period along with a corresponding *net position* (calculated as the difference between the matched supply and the matched demand quantities belonging to that *bidding zone*).

Bidding zones can exchange energy between them in an ATC model (Section 4.2), a flow based model (Section 4.3) or a hybrid model (combination of the previous two models).

The *net position* of a *bidding zone* can be subject to limitations in the variation between periods.

4.2.1. Net position ramping (periodic and daily)

The algorithm supports the limitation on the variations of the *net position* from one period to the next. The periods relate to the MTU of the bidding zone. There are two ramping requirements that can be imposed on the *net position*.

- Periodic *net position* ramping: this is a limit on the variation of the *net position* of a *bidding zone* from one period to the next.
- Daily (or cumulative) *net position* ramping: this is a limit on the amount of reserve capacity that can be used during the day.

Reserve capacity is needed as soon as the variation of the *net position* from one period to the next exceeds a certain threshold. There is a fixed limit on the total amount of reserve that can be used during the day. Reserve capacity is defined separately for each direction (increase/decrease).
By including the *net position* of the last period for the previous (delivery) day, overnight ramping can be taken into account.

4.3. ATC Model

In an ATC model, the *bidding zones* are linked by interconnectors (bidding zone lines) representing a given topology. The energy from one *bidding zone* to its neighbouring zone can only flow through these lines and is limited by the available transfer capacity (ATC) (Section 4.3.1) of the line.

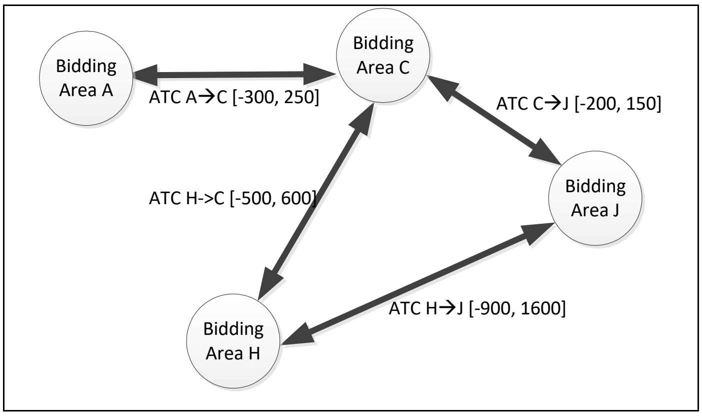


Figure 2 – *Bidding zones* connected in ATC model

Additional restrictions may apply to the interconnectors:

- The flow through a line can be subject to losses (Section 4.3.2)
- The flow through a line can be subject to tariffs (Section 4.3.3)
- The flow variation between two consecutive periods can be restricted by an periodic flow ramping limit (Sections 4.3.4 and 4.3.5)

4.3.1. Available Transfer Capacity (ATC)

ATC limitations constrain the flow that passes through the interconnectors of a given topology.

In EUPHEMIA, lines are oriented from a source *bidding zone* (A) to a sink *bidding zone* (C). Thus, in the examples hereafter, a positive value of flow on the line indicates a flow from A to C, whereas a negative value indicates a flow from C to A.

The available transfer capacity of a line can be different per period and direction of the line (Figure 2).

- As an example, let us consider two *bidding zones* A and C connected by a single line defined from A to C (A→C). For a given period, the ATC in the direction (A→C) is assumed to be equal to 250 MW and equal to 300 MW in the opposite direction (C→A). In practice, this implies that the valid value for the algebraic flow through this line in this period shall remain in the interval [-300, 250].

ATC limitations can also be negative. A negative ATC value in the same direction of the definition of the line A→C (respectively, in the opposite direction C→A) is implicitly indicating that the flow is forced to only go in the direction C→A (respectively, A→C).

- In the previous example, if the ATC was defined to be equal to -250 MW instead of 250 MW in the direction A→C then this would imply that the valid value for the flow will now be in the interval [-300, -250], forcing the flow to be in the C→A direction (negative values of the flow on a line defined as A→C).

4.3.2. Losses

Flow through a line between *bidding zones* may be subject to losses. In this case, part of the energy that is injected in one side of the line is lost, and the energy received at the end of the cable is less than the energy initially sent (Figure 3).

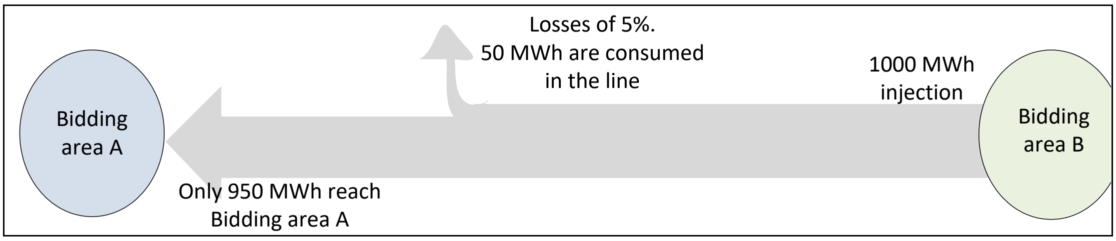


Figure 3 – Example of the effect of losses in one line.

4.3.3. Tariffs

In an ATC network model, the DC cables might be operated by merchant companies, who levy the cost incurred for each 1MWh passing through the cable. In the algorithm, these costs can be represented as flow tariffs.

The flow tariff is included as a loss with regard to the *congestion rent*. This will show up in the results as a threshold for the price between the connected bidding zones. If the difference between the two corresponding market clearing prices is less than the tariff then the flow will be zero. If there is a flow the price difference will be exactly the flow tariff, unless there is congestion. Once the price difference exceeds the tariff the *congestion rent* becomes positive.

4.3.4. Periodic Flow Ramping Limit on Individual Lines

The periodic variation of the flows through an interconnector can be constrained by a ramping limit. This limitation confines the flow in an "allowed band" when moving from one period to the next (Figure 4). The ramping limit constrains the flow that can pass through the line in period t depending on the flow that is passing in the previous period t--1.

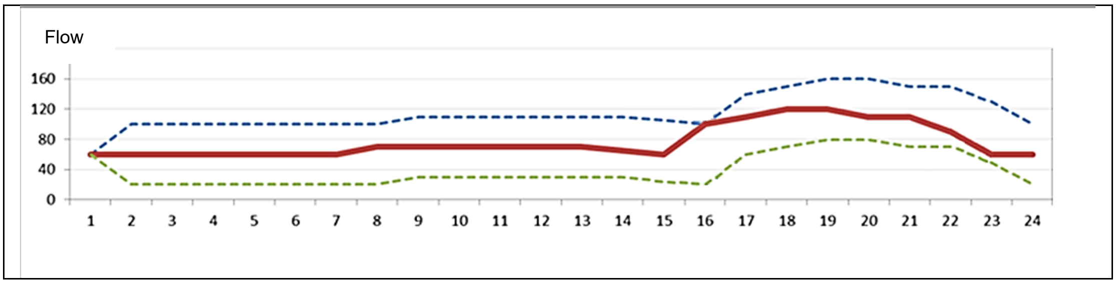


Figure 4 – Effect of the periodic flow ramping limit. The flow stays in the allowed band between periods.

The ramping limit is defined by: The maximum increment of flow from period t-1 to period t (called ramping-up), and the maximum decrement of flow from period t-1 to period t (called ramping-down). The ramping limits may be different for each period and direction. For period 1, the limitation of flow takes into account the value of the flow of the last period of the previous day.

4.3.5. Constraints on Line Sets

4.3.6. Periodic Flow Ramping Limit on Line Sets

Flow ramping constraints can apply to a group of interconnectors at once, i.e. the sum of the flows through a set of lines can be restricted by ramping limits.

- As an example, let us consider a line set composed by two interconnectors: the former between areas A and B the latter between areas A and C. If we set the periodic flow ramping limit for this line set to 450 MW, this will enforce that the sum of the flow from bidding zone A to B and the flow from bidding zone A to C is allowed to vary by only 450 MW from one period to the next.

4.3.7. Line set capacity constraint

Cumulative capacity constraints can apply to a group of interconnectors at once, i.e. the sum of the flows through a set of lines can be restricted by cumulative capacity limits.

- As an example, let us consider a line set composed by two interconnectors: the former between areas A and B and the latter between areas A and C. If we set the cumulative capacity for this line set to 1000 MW, this will enforce that the sum of the flow from bidding zone A to B and the flow from bidding zone A to C cannot exceed 1000 MW.

4.3.8. Parallel ATCs

Implementation of the parallel ATCs functionality gives the possibility to model several lines connecting the same pair of bidding zones.

Since bi-directional flows are allowed among these distinct parallel lines, it is acknowledged by TSOs that this implementation could possibly induce loops in case of negative prices and losses. It is also acknowledged that with positive prices the line with lower losses will be prioritized. The opposite is true with negative prices, in which case the line with higher losses will be prioritized.

Flow indeterminacy between parallel lines is managed using linear and quadratic cost coefficients.

Parallel lines are also propagated to SA and NTH levels, i.e. parallel ATC lines connecting two bidding zones shall also connect the corresponding SAs and NTHs of these bidding zones. If parallel ATC lines exist between two bidding zones, the flow on a SA line between these two bidding zones is calculated for each of these ATC lines. The same applies for NTH lines.

The functionality allowing the forwarding of residuals to "parent areas" will not be supported together with the use of parallel lines. I.e. this option to allocate rounding residuals from virtual areas to their parent area would not be supported where the virtual area is connected to its parent by more than one line.

4.3.9. External constraint

An external constraint when applied enforces global import or export limits for each bidding zone and for each period.

In EUPHEMIA the $NET\_POSITION_{m,t}$ of a bidding zone *m* and a period h represents the total net exchange of this bidding zone with its neighbors, both through the flow-based and ATC network models.

External constraint defines new bounds to this $NET\_POSITION_{m,t}$ variable, to limit the global import/export of each bidding zone at each period:

$$NET\_POSITION_{m,t} \leq MAX\_EXPORT_{m,t}$$
$$-MAX\_IMPORT_{m,t} \leq NET\_POSITION_{m,t}$$

The bounds need to be positive numbers and coherent with the optional ramping requirement in order to avoid infeasibilities in the primal problem.

4.4. Flow Based Model

The Flow Based (FB) model is an alternative to ATC network constraints. Modeling network constraints using the flow- based model allows a more precise modeling of the physical flows.

The FB constraints are given by means of two components:

- **Remaining Available Margin (RAM):** number of MW available for exchanges
- **Power Transfer Distribution Factor (PTDF):** ratio which indicates how much MWh are used by the *net position*s resulting from the exchanges

PTDFs can model different network constraints that constrain the exchanges allowed. Each constraint corresponds to a single row in the $PTDF$ matrix, and has one corresponding margin (one value of the RAM vector). The $PTDF$ matrix has columns for each hub where it applies to (e.g. FB in CWE has columns for the *net position*s of all CWE hubs: BE, DE, FR and NL). Net position in the FB context should be read as the net position of a market as a result of the exchanges via the meshed (flow-based) network (thus excluding the exchange via ATC lines).

Therefore, the constraint that is being imposed is the following:

$$PTDF \cdot nex \leq RAM$$

Here nex is the vector of *net position*s which are subject to the flow based constraints. The flow based modeling has some consequences to price formation, and can potentially result in “non-intuitive” situations that happen when the energy goes from high priced areas to low priced areas.

Example:
Consider a three market example (Figure 5), with a single PTDF constraint:

$$0.25 \cdot nex_A - 0.5 \cdot nex_B - 0.25 \cdot nex_C \leq 125$$

And consider the market outcome shown in Figure 5 below.

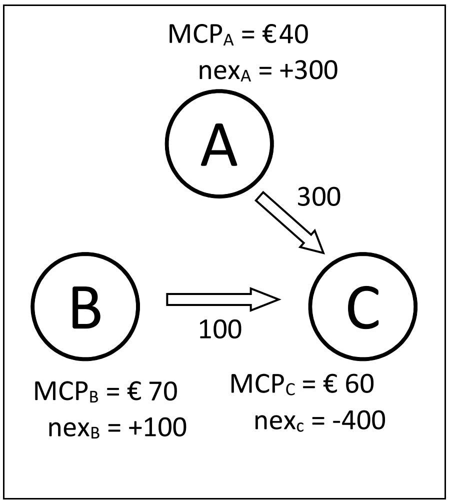


Figure 5 – Example of *net position*s decompositions into flows

In the representation of the result, "bilateral exchanges" between *bidding zones* have been indicated. This is merely one potential decomposition of *net position*s into flows out of many. Alternative flows could have been reconstructed too. However, since market B is exporting energy, whereas it is the most expensive market, any breakdown into flows shall result in market B exporting energy to a cheaper market.

Non-Intuitiveness
From the example above we see that FB market coupling can lead to non-intuitive situations. The reason is that some non-intuitive exchanges free up capacity, allowing even larger exchanges between other markets. In our example, exporting from B to C loads the critical branch with (-0.5) – (-0.25) = -0.25 MWh for each MWh exchanged, i.e. it actually relieves the line. Economic surplus maximization can therefore lead to these non-intuitive situations.

Multi time resolutions
In this section we have not made any reference to any time period, suggesting the FB constraints for the different periods of the day are independent, and there exists only a single time resolution for all constraints. In the next section on LTA inclusion we report the impacts of unharmonized time resolutions, and refer to the annex where details are disclosed.

Flow-factor competition at maximum price
Another side-effect of the Flow-based model is the flow factor competition in case of market curtailment at maximum price. If several markets end up at maximum price in a flow-based domain, the PTDF coefficients can lead

to unfair distribution of the available energy and in some extreme cases, the solution that maximizes the economic surplus is the one where one market is totally curtailed while all the available energy is given to another market which is not necessarily at maximum price. EUPHEMIA implements a mechanism that allows a fairer distribution of the curtailment between all the markets in a Flow-based domain.

4.4.1. Extended LTA inclusion

Apart from the regular Flow Based model, EUPHEMIA can also be configured to manage the LTA (Long Term Allocation) inclusion:

The LTA domain includes the long-term capacities allocated explicitly which are offered for some borders. If the capacities are in the form of FTRs (financial transmission rights), or they are not nominated PTRs (physical transmission rights), where UIOSI (use-it-or-sell-it) applies, the LTA creates financial obligations for the TSOs. This should not be an issue, as the TSOs hedge for these financial obligations with the actual capacities allocated in the day-ahead coupling. The congestion rent, which is a result of the day-ahead capacity allocation, should cover their obligations.

An issue emerges when the Flow Based model is used for the day-ahead allocation: FB capacity calculation considers the best available forecast of the grid constraints and delivers a FB domain respecting them. In turn this may result in a FB domain not covering the LTA domain, which may result in congestion rent being insufficient to cover the obligations coming from the LT allocated capacities.

To mitigate this risk, CWE FB implementation introduced the concept of LTA inclusion, where the FB domain would be extended to always cover the LTA domain. To still only impose convex constraints on the EUPHEMIA algorithm, this was implemented by effectively computing the convex hull between the original FB domain and the LTA domain[1].

Unfortunately, this approach does not scale well: when CWE FB was first introduced in BE, DE/AT/LU, FR and NL, TSOs submitted an average number around 630 constraints per day, which went up to around 3 500 per day when DE and AT were split. This one additional borders added relatively few extra constraints, but the convex hull required many additional virtual constraints.

With the integration of the Alegro interconnector between BE and DE in CWE FB, a further increase in virtual constraints is anticipated. Preliminary figures suggested to expect as many as 27 000 constraints per day, which would raise concerns for EUPHEMIA scalability.

To mitigate the scalability issue, in EUPHEMIA 10.5 extended LTA inclusion is supported. Rather than letting TSOs extend the FB domain a-priori, instead the original ("virgin") FB domain is provided, as well as the LTA domain that shall be covered. The EUPHEMIA model now has to deal with some additional complexity to consider the LTA domain, and effectively respect the same constraints as would have come from the pre-extended domain. However, the few extra constraints and variables mean that rather than facing 27 000 FB constraints, only the actual FB constraints need to be considered, which preliminary data suggests are as few as 790, i.e. fewer than after the DE/AT split, and closer to the period before the split.

[1] See annex 16_6 information regarding LTA inclusion from https://www.jao.eu/support/resourcecenter/overview?parameters=%7B%22IsCWEFBMCRelevantDocumentation%22%3A%22True%22%7D

For the special case where the LTA domain is empty, and only the virgin FB domain is considered, this is equivalent to the FB network constraints mentioned in the previous section.

15’ MTU
Since a FB model spans several bidding zones, it may not necessarily be the case that all bidding zones support the same time resolution. Even if they did, Euphemia does not dictate that the LTA lines connecting the different bidding zones have a harmonised time resolution. The only constraint imposed is that the bidding zones and lines have a time resolution no finer than that of the FB balancing area. So on the side of the capacity calculation, any configuration of input data is supported.

However, for the allocation of the capacity Euphemia considers the time resolution of the balancing area. If a line has a coarser time resolution than that of the FB region, the LTA may be allocated at the time resolution of the FB region.
Example:
Imagine a 15’ FB region, that includes a 30’ line. For some 30’ period the LTA = 100 for this line.
Euphemia may allocate 30MW for the first 15’ sub-period and 100MW for the second 15’ sub-period.

Mind that this logic is not extended to the level of the bidding zone: the net position of a bidding zone shall necessarily follow its time resolution.

To better understand the details of the capacity allocation with potentially unharmonized time resolution, the reader is advised to consult the section on Extended LTA Inclusion in Annex C Mathematical Approach.

4.5. Scheduling Area Topology

4.5.1. Scheduling Areas

Scheduling areas define a sub-level of bidding zones: one or more scheduling areas must be present in each bidding zone, and aim at modeling scheduling exchanges in bidding zones where several TSOs coexist.

Unlike bidding zones, scheduling area net positions cannot themselves be subject to limitations.

4.5.2. Scheduling Area Lines

Scheduling areas can exchange energy between them through *Scheduling Area Lines*. These lines may connect scheduling areas within a same bidding zone, or scheduling areas corresponding to distinct bidding zones (in the latter case, a line between the two corresponding bidding zones must exist). One or more scheduling area lines may be associated to a line between two bidding zones.

Scheduling area lines are populated with so-called *Thermal Capacities*. These values do not in themselves bound the energy exchanges between scheduling areas. They are however used to uniformly distribute energy between a set of scheduling area lines in case several of them are associated with a same bidding zone line. See section 7.9.5 for more details.

If multiple scheduling areas exist within a given bidding zone, they shall all be (directly or indirectly) connected to each other so that a unique price can be determined by EUPHEMIA.

4.6. NEMO Trading Hub Topology

4.6.1. NEMO Trading Hubs

Orders cannot directly be submitted in bidding zones, nor scheduling areas. They are associated to *NEMO Trading Hubs (NTHs)*. In each Scheduling Area, there shall exist (unless specific exceptions) one or more NEMO trading hubs.

NEMO trading hub net positions cannot be subject to limitations.

4.6.2. NEMO Trading Hub lines

NEMO trading hubs can exchange energy between them through *NEMO Trading Hub Lines*. These lines may connect NTHs within a same scheduling area, or NTHs corresponding to distinct scheduling areas (in the latter case, a line between the two corresponding scheduling areas must exist). One or more NTH lines may be associated to a line between two scheduling areas.

NTH lines are not provided with any specific property: any capacity may transit between two NTHs. Also, all NTHs of a same scheduling area shall be (directly or indirectly) connected so that EUPHEMIA can determine a unique price. See section 7.9.5 for more details.

5. Market Orders

The algorithm can handle a large variety of order types at the same time, which are available to the market participants in accordance with the local market rules:

- Aggregated Periodic Orders
- Complex Orders
    - MIC orders
    - Load Gradient orders
- Block Orders

  - Linked Block Orders
  - Exclusive Groups of Block Orders
  - Flexible Orders
- Merit Orders and PUN Orders.

5.1. Aggregated Period Orders

Demand (resp. supply) orders from all market participants belonging to the same *bidding zone* will be aggregated into a single curve referred to as aggregated demand (resp. supply) curve defined for different periods. These periods may be 15', 30' or 60' periods. Demand orders are sorted from the highest price to the lowest. Conversely, supply orders are sorted from the lowest to the highest price.

Aggregated supply and demand curves can be of the following types:

- Linear piecewise curves containing only interpolated orders (i.e. two consecutive points of the monotonous curve cannot have the same price, except for the first two points defined at the maximum / minimum prices of the *bidding zone*).

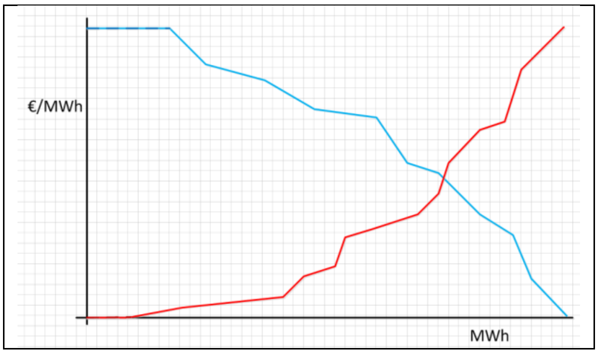


Figure 6 – Linear piecewise aggregated curve.

- Stepwise curves containing only step orders (i.e. two consecutive points always have either the same price or the same quantity).

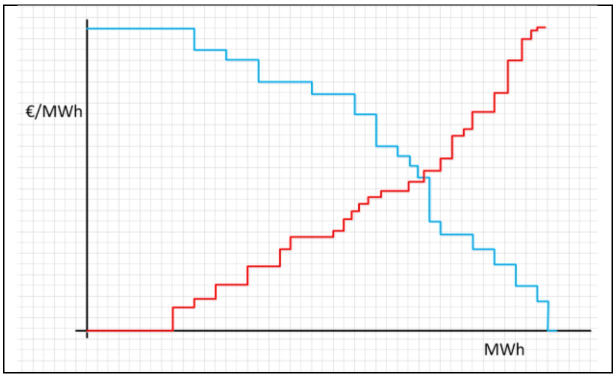


Figure 7 – Stepwise aggregated curve.

- Hybrid curves containing both types of orders (composed by both linear and stepwise segments).

The following nomenclature is used when speaking about period orders and market clearing prices:

- One demand (resp. supply) period order is said to be *in-the-money* when the arithmetic mean of the market clearing price(s) of the MTU(s) contained in the period is lower (resp. higher) than the price of the curve order.
- One demand or supply period order is said to be *at-the-money* when the price of the curve order is equal to the arithmetic mean of the market clearing price(s) of the MTU(s) contained in the period.
- One demand (resp. supply) period order is said to be *out-of-the-money* when the arithmetic mean of the market clearing price(s) of the MTU(s) contained in the period is higher (resp. lower) than the price of the curve order.
- For linear piecewise period orders starting at price p_0 and finishing at price p_1, p_0 is used as the order price for the nomenclature above (except for energy *at-the-money*, where the arithmetic mean of the market clearing price(s) of the MTU(s) contained in the period is in the interval $[p_0, p_1]$).

The rules that apply for the acceptance of period orders in the algorithm are the following:

- Any order submitted at the time resolution of the MTU of the bidding zone that is in-the-money must be fully accepted.
- Any order out-of-the money must be rejected.
- Orders at-the-money can be either accepted (fully or partially) or rejected.
- Orders submitted at a time resolution that is coarser than the MTU of the bidding zone they belong to are allowed to be paradoxically rejected.

Price-taking orders, defined at the maximum / minimum prices of the *bidding zone*, have additional requirements which are detailed in Section 6.5.1.

5.2. Complex Orders

A complex order is a set of simple supply stepwise curve orders (which are referred to as curve sub-orders) belonging to a single market participant, spreading out along different periods and are subject to a complex condition that affects the set of curve sub-orders as a whole.

A complex order can be a sell or buy order.

A complex order is composed of:

- Curve orders, one set per period, expressed in the same time resolution of the bidding zone where they are submitted.

- A complex order is defined with one time resolution and the curve sub-orders are offered with the same time resolution defined for the complex order.
- All sub-orders should be of the same type (sell or buy) defined for the complex order.

- Additional complex conditions:
  - MIC condition / MP condition
    - MIC condition is the Minimum Income Condition and can be defined for sell complex orders.
    - MP condition is the Maximum Payment condition and can be defined for buy complex orders.
  - Load Gradient condition
  - A combination of MIC condition / MP condition and load gradient condition

When a complex order makes use exclusively of MIC/MP condition, then it can be referred as "pure MIC/MP order", whereas a complex order that makes use exclusively of load gradient condition , it can be referred as "pure Load Gradient order".

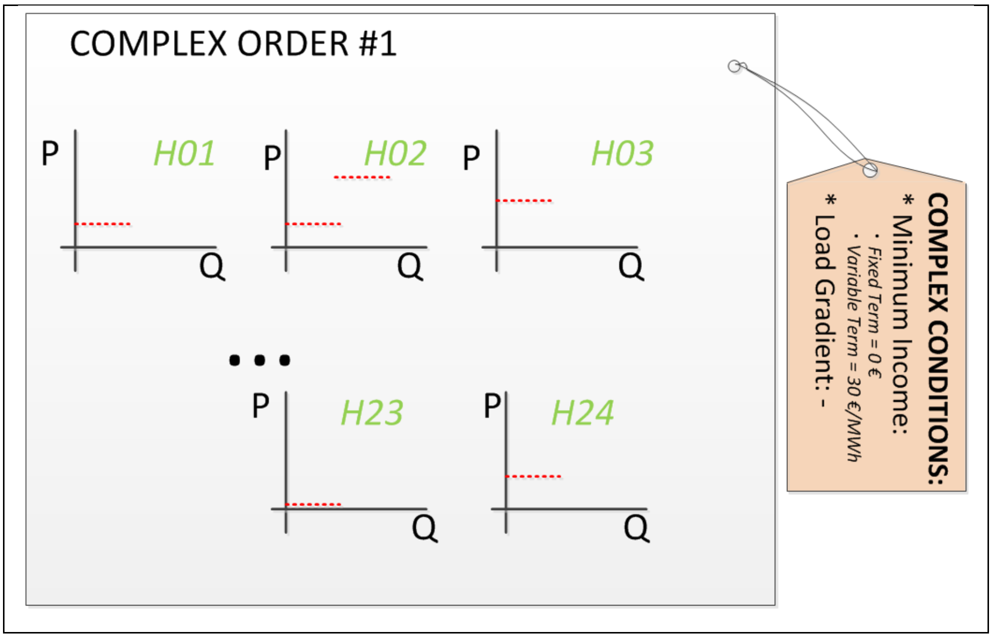


Figure 8 – A complex order is composed of a set of curve sub-orders (in dotted line) associated with complex conditions. This example uses hourly periods, numbered from hour 1 to hour 24. For a 30' complex orders they would number 1 to 48, for a 15' MTU complex order they would number 1 to 96.

Since several NEMOs can be present in the same bidding zone, complex orders of NEMOs that belong to the same bidding zone need to be combined. Complex orders' IDs uniqueness within one bidding zone will be assured by generating unique internal complex order IDs per session automatically.

Furthermore, each complex order will also be associated with a hash: this hash can then be used for settling ties between identical complex orders submitted by different NEMOs in the same bidding zone. More information is available in paragraph 5.2.4

5.2.1. Minimum Income Condition (MIC) / Maximum Payment condition (MP) in complex orders

The Minimum Income Condition (**MIC**) (respectively Maximum Payment condition (**MP**)) in complex orders adds an economic condition to sell complex order (respectively, buy complex order), which represents the minimum income (respectively, the maximum payment) expected, by order's owner defined by a fix term in euros or/and a variable term in euros per accepted MW produced (consumed, respectively) for the set of curve sub-orders.

Generally speaking, the MIC constraint means that the amount of money collected by the order in all periods must cover its production costs, which is defined by a fix term (for a MIC it's representing the startup cost of a power plant) and a variable term multiplied by the total assigned energy (for MICs it's representing the operation cost per produced MW for a period in a power plant).

For the case of MP constraint, it means that the amount of money to be paid by the order in all periods must be less or equal than the maximum amount of payment that the order is willing to do for the energy consumed, which is defined by a fix term and a variable term multiplied by the total assigned energy.

The MIC condition (respectively, MP condition) constraint is defined by:

- A fix term (FT) in Euros
- A variable term (VT) in Euros per accepted MW.

In the final solution, MIC orders are activated or deactivated (as a whole):

- If the economic condition is not fulfilled, the complex order having MIC condition (respectively, MP condition) must be rejected.
  - In this case, each of the curve sub-orders of the MIC/MP are fully rejected, even if it is in-the-money (with the exception of scheduled stop for MIC orders, see Section 5.2.2).
- If the economic condition is fulfilled, the complex order having MIC condition (respectively, MP condition) can be accepted.
- If the economic condition is fulfilled, but the complex order having MIC condition (respectively, MP condition) order is rejected, the complex order having MIC condition (respectively, MP condition) is then defined as paradoxically rejected.
- The final solution given by EUPHEMIA will not contain active MIC orders (respectively, MP orders) not fulfilling their Minimum Income Condition (respectively Maximum Payment) constraint. There orders are also known as paradoxically accepted MICs (respectively paradoxically accepted MPs).

5.2.2. Scheduled Stop in complex orders

In case the owner of a power plant which was running the previous day offers a MIC order to the market, he may not want to have the production unit stopped abruptly in case the MIC is deactivated.

For the avoidance of this situation, the sender of a MIC has the possibility to define a "scheduled stop". Using a schedule stop will alter the deactivation of the MIC: the deactivation will not imply the automatic rejection of all the curve sub-orders. On the contrary, the first (i.e. the cheapest) curve sub-order in the periods that contain scheduled stop will not be rejected but will be treated as any curve order.
Scheduled stop periods must be consecutive, can start on the first period of the day and can extend up to the 3 first hours of the day. Hence if a complex order has a time resolution of 60/30/15 min, it cannot declare more than 3/6/12 scheduled stop periods respectively.

No scheduled stop may be defined for MP complex orders.

5.2.3. Load Gradient in complex orders

Complex orders (with their set of curve sub-orders) on which a Load Gradient constraint applies are called Load Gradient Orders.

Generally speaking, the Load Gradient constraint means that the amount of energy that is matched by the curve sub-orders belonging to a Load Gradient order in one period is limited by the amount of energy that was matched by the curve sub-orders in the previous period. There is a maximum increment / decrement allowed (the same value for all periods). Period 1 is not constrained by the energy matched in the last period of the previous day. If only one of these values is defined, the other value (i.e. empty) is considered as unconstrained.

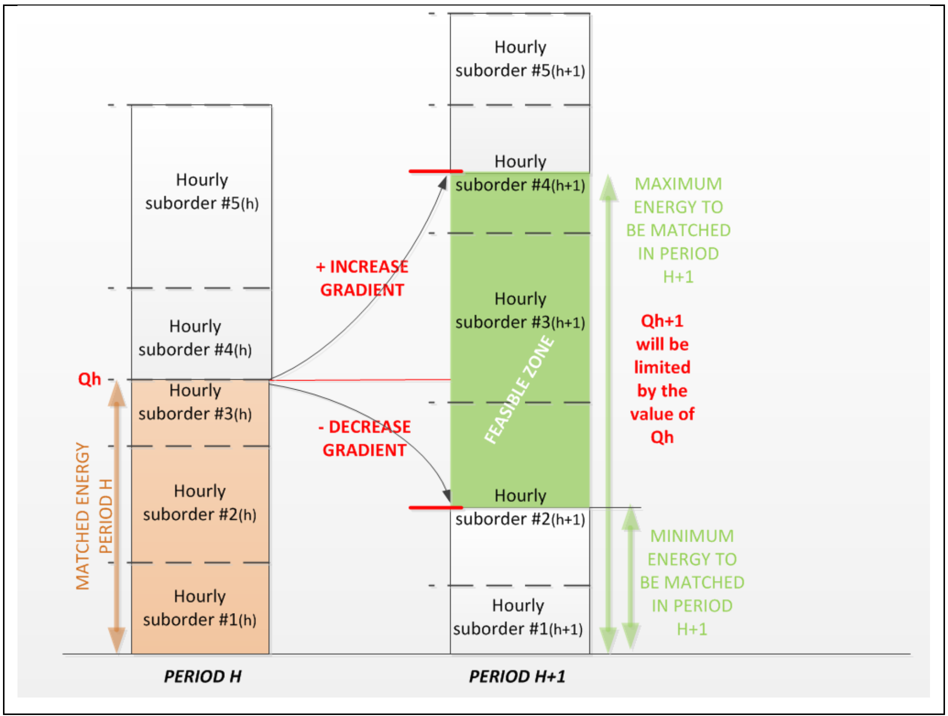


Figure 9 – A Load Gradient order. Effect produced by the amount that is matched in period (h) on period (h+1). The example shows an hourly Load Gradient order. Analogously 30' and 15' examples can be imagined with 30' respectively 15' gradients.

5.2.4. Complex order tie rules

EUPHEMIA implements complex order tie rules to arbitrate between identical complex orders in the same bidding zone, when only some, but not all can be activated in the final solution.

Two complex orders are considered equal, if:

- The time resolutions are identical;
- The bidding zones are identical;
- The signs (buy or sell) are identical;
- The fixed terms are identical;
- The variable terms are identical;
- The increase gradients are identical;
- The decrease gradients are identical;
- The scheduled stop periods are identical;
- The sub orders have identical:
    - Periods;
    - Prices;
    - Powers;

For this case, economic criteria are insufficient to arbitrate: accepting one or the other will result in identical economic surplus. Instead, some secondary criteria are used to make the arbitration, and allow ties to be deterministically broken:

1. The complex order with an earlier last modification timestamp will be prioritized;
2. If 1. does not break the tie, we consider two sub cases:
    a. For bidding zones where only a single NEMO exists, the priority is set according to the lowest "external id", the id assigned to the complex order by the local trading system of the corresponding power exchange. These ids must be unique, and therefore will necessarily break any tie;
    b. For bidding zones with multiple NEMOs ties are broken differently: To avoid unequal treatment the preferred complex order is selected "randomly": random in the sense bias are avoided, and complex orders from one NTH will not be more or less likely to be accepted than complex orders from another NTH. In order to make sure EUPHEMIA behaviour is repeatable, repeatable randomness is applied. This is managed by using the hashes that were compiled for each complex order (on the basis of the different parameters describing the complex orders). These hashes will be used to settle ties, and should be sufficiently random to meet this fairness objective.

5.3. Scalable Complex Orders

A Scalable complex order is a set of stepwise curve orders (which are referred to as curve sub-orders) belonging to a single market participant, spreading out along different periods and are subject to an economic condition that affects the set of curve sub-orders as a whole.

A scalable complex order can be a sell or buy order

A Scalable complex order (or SCO) is composed of:

- A set of stepwise curve sub-orders (sell for scalable MIC orders; buy for scalable MP orders), one set per period in the same MTU resolution of the bidding zone they are submitted.
    - A scalable complex order is defined with one time resolution and the curve sub-orders are offered with the same time resolution defined for the scalable complex order.
    - All sub-orders should be of the same type (sell or buy) defined for the scalable complex order.
- A minimum acceptance power, one value per period, which will be 0 if not provided.
- Additional conditions:
    - Scalable MIC condition / scalable MP condition:
        - Scalable MIC condition can be defined for sell scalable complex orders

  - Scalable MP condition can be defined for buy scalable complex orders
- Load gradient condition
- A combination of scalable MIC condition / MP condition and load gradient condition.

5.3.1. Minimum Income Condition (MIC) / Maximum Payment condition (MP) in scalable complex orders

The Minimum Income Condition (**MIC**) (respectively Maximum Payment condition (**MP**)) in scalable complex orders adds an economic condition to sell scalable complex order (respectively, buy scalable complex order), which represents the minimum income (respectively, the maximum payment) expected, by order's owner defined by a fix term in euros produced (consumed, respectively) for the set of curve sub-orders.

Generally speaking, the MIC constraint means that the amount of money collected by the order in all periods must cover its production costs, which is defined by a fix term (for a MIC it's representing the startup cost of a power plant) and the steps of the set of stepwise curve sub-orders in all periods.

For the case of MP constraint, it means that the amount of money to be paid by the order in all periods must be less or equal than the maximum amount of payment that the order is willing to do for the energy consumed, which is defined by a fix term and the steps of the set of stepwise curve sub-orders in all periods.

The MIC condition (respectively, MP condition) constraint in scalable complex orders is defined by:

- A fix term (FT) in Euros
- The steps of the set of stepwise curve sub-orders in all periods.

In the final solution, SCO orders are activated or deactivated (as a whole):

- If the MIC/MP economic condition is not fulfilled, the scalable complex order must be rejected.
  - In this case, each of the curve sub-orders of the MIC/MP is fully rejected, even if it is in-the-money (with the exception of scheduled stop for MIC condition in scalable complex orders, see Section 5.3.2)
- If the MIC/MP economic condition is fulfilled, the scalable complex order can be accepted.
- If the MIC/MP economic condition is fulfilled but the scalable complex order is rejected, the scalable complex order is then defined as paradoxically rejected.

Additionally, Scalable complex orders cannot be accepted for a power less than the minimum acceptance power defined for all and each one of the periods.

The final solution given by EUPHEMIA will not contain active MIC orders (respectively, MP orders) not fulfilling their Minimum Income Condition (respectively Maximum Payment) economic condition. There orders are also known as paradoxically accepted MICs (respectively paradoxically accepted MPs).

5.3.2. Scheduled Stop in scalable complex orders

The scheduled stop in scalable complex orders works the same than with complex orders.

In case the owner of a power plant which was running the previous day offers a MIC order to the market, he may not want to have the production unit stopped abruptly in case the MIC is deactivated.

For the avoidance of this situation, the sender of a MIC has the possibility to define a “scheduled stop”. Using a schedule stop will alter the deactivation of the MIC: the deactivation will not imply the automatic rejection of all the curve sub-orders. On the contrary, the first (i.e. the cheapest) curve sub-order in the periods that contain scheduled stop will not be rejected but will be treated as any curve order.

Scheduled stop periods must be consecutive, can start on the first period of the day and can extend up to the 3 first hours of the day. Hence if a scalable complex order has a time resolution of 60/30/15 min, it cannot declare more than 3/6/12 scheduled stop periods respectively.

No scheduled stop may be defined for demand scalable complex orders.

5.3.3. Load Gradient in scalable complex orders

The load gradient in scalable complex orders works the same than with complex orders.

Scalable complex orders (with their set of curve sub-orders) on which a Load Gradient constraint applies are called Load Gradient Scalable Orders.

Generally speaking, the Scalable Load Gradient constraint means that the amount of energy that is matched by the curve sub-orders belonging to a Load Gradient order in one period is limited by the amount of energy that was matched by the curve sub-orders in the previous period. There is a maximum increment / decrement allowed (the same value for all periods). Period 1 is not constrained by the energy matched in the last period of the previous day. If only one of these values is defined, the other value (i.e. empty) is considered as unconstrained.

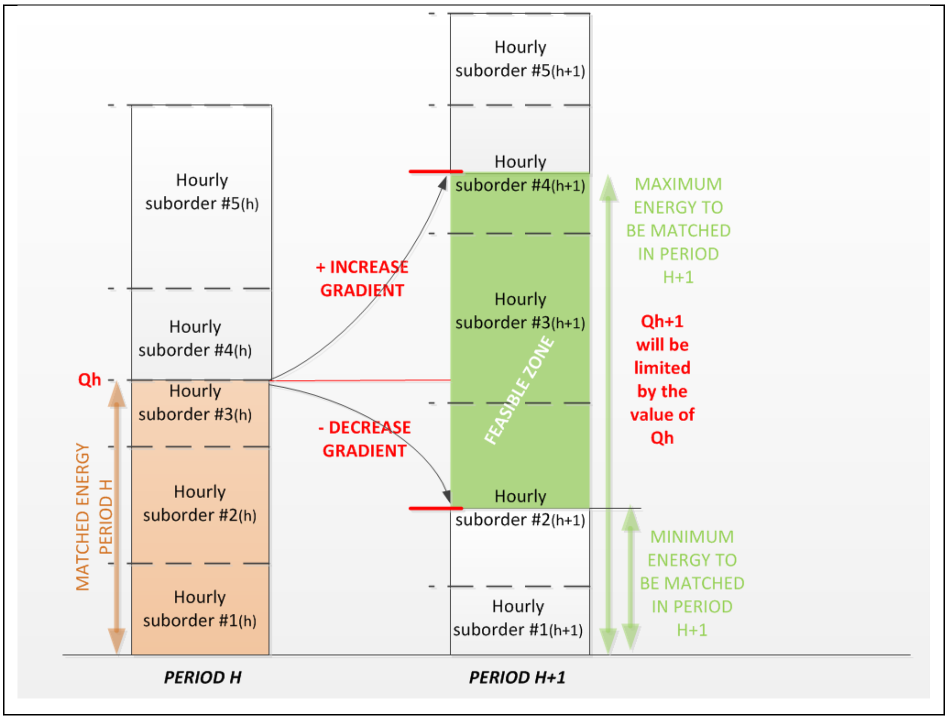


Figure 10 – A Load Gradient order. Effect produced by the amount that is matched in period (h) on period (h+1).

5.3.4. Scalable complex order tie rules

The tie rules in scalable complex orders works the same than with complex orders, with the difference that the tie-break rule does not take the variable term into account since it is not part of the definition of the scalable complex orders. However, the minimum acceptance powers in each period have to be the same as well to consider two scalable complex orders identical.

EUPHEMIA implements scalable complex order tie rules to arbitrate between identical scalable complex orders in the same bidding zone, when only some, but not all can be activated in the final solution.

Two scalable complex orders are considered equal, if:

- The time resolutions are identical
- The bidding zones are identical;
- The signs (buy or sell) are identical;
- The fixed terms are identical;
- The increase gradients are identical;
- The decrease gradients are identical;
- The scheduled stop periods are identical;
- The minimum acceptance powers are identical in all periods;
- The sub orders have identical:
    - Periods;
    - Prices;
    - Powers;

For this case, economic criteria are insufficient to arbitrate: accepting one or the other will result in identical economic surplus. Instead, some secondary criteria are used to make the arbitration, and allow ties to be deterministically broken:

1. The scalable complex order with an earlier last modification timestamp will be prioritized;
2. If 1. does not break the tie, we consider two sub cases:
    a. For bidding zones where only a single NEMO exists, the priority is set according to the lowest "external id", the id assigned to the scalable complex order by the local trading system of the corresponding power exchange. These ids must be unique, and therefore will necessarily break any tie;
    b. For bidding zones with multiple NEMOs ties are broken differently: To avoid unequal treatment the preferred scalable complex order is selected "randomly": random in the sense bias are avoided, and scalable complex orders from one NTH will not be more or less likely to be accepted than scalable complex orders from another NTH. In order to make sure EUPHEMIA behaviour is repeatable, repeatable randomness is applied. This is managed by using the hashes that were compiled for each scalable complex order (on the basis of the different parameters describing the scalable complex orders). These hashes will be used to settle ties, and should be sufficiently random to meet this fairness objective.

5.4. Block Orders

A block order is defined by:

- sense (supply or demand)
- price limit (minimum price for supply block orders and maximum price for demand block orders),
- periods contained by the block
- volume that can be different for every period
- minimum acceptance ratio.

In the simplest case, a block order is defined for a consecutive set of periods with the same volume and with a minimum acceptance ratio of 1. These are usually called regular (fill-or-kill) block orders. In general, the periods of the block orders can be non-consecutive, the volume can differ over the periods and the minimum acceptance ratio can be less than 1 (Curtailable Block Orders –partial acceptance is allowed).

Example of a block order:

Block Order #1

- Sense: supply
- Price: 40 €/MWh
- Minimum acceptance ratio: 0.5
- Intervals: Periods (3-7), periods (8-19) and periods (22-24)
- Volume: 80 MWh in the first interval, 220 MWh in the second one, and 40 MWh in the third one.

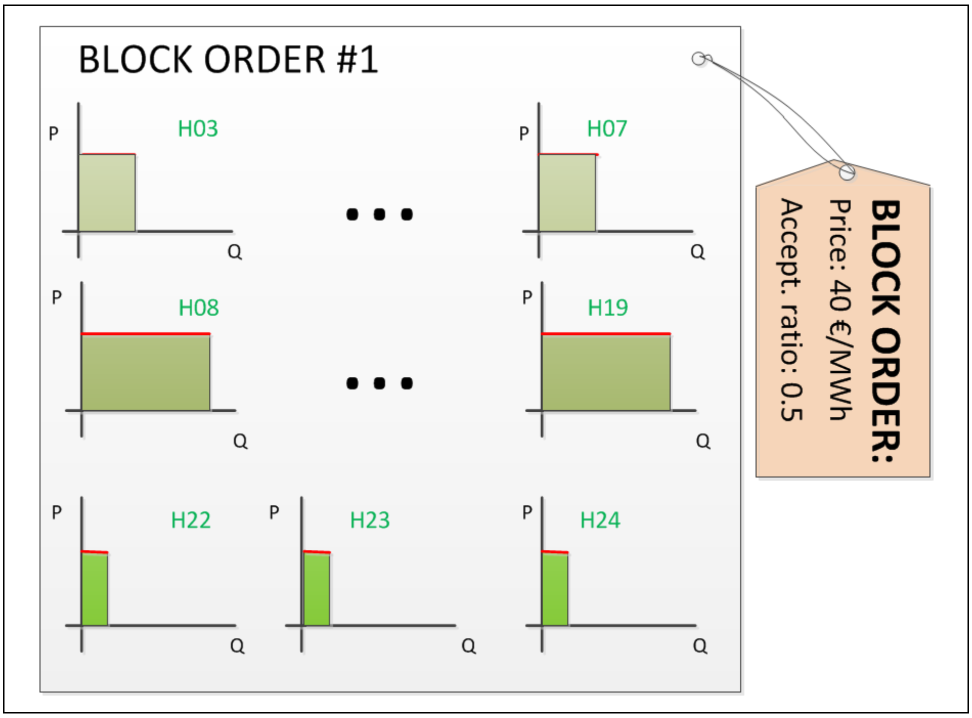


Figure 11 – Block order example (the example shows a 60’ or hourly block, but Euphemia also supports 15’ and 30’ blocks)

Block orders can have a time resolution equal to, or coarser than that NEMO trading hub resolution to which they are submitted. The block profile is defined via intervals whose period indices are compliant with the block time resolution.

Block orders that are *out-of-the-money* cannot be accepted. As a consequence, all block orders will fall in one of the below categories:

- if the block is *in-the-money* or *at-the-money*, then the block can be one of: fully rejected (PRB), entirely accepted or partially accepted (PPRB), to the extent that the ratio "accepted volume/total submitted volume" is greater than or equal to the minimum acceptance ratio of the block (e.g. 0.5) and equal over all periods;
- or if the block is *out-of-the-money*, then the block must be entirely rejected;

Block orders have a single acceptance ratio that applies to the full block profile. If a block that spans more than 1 period is (partially) accepted, the accepted quantity in each period is this ratio multiplied by submitted quantity for each period the block spans rounded to the nearest volume tick of the market.

Since several NEMOs can be present in the same bidding zone, block orders of NEMOs that belong to the same bidding zone need to be combined, despite their order type ("normal" blocks, linked block families, flexible orders and exclusive groups).
Block IDs' uniqueness within one bidding zone will be assured by generating unique internal block IDs per session automatically.

Furthermore, each block will also be associated with a hash: this can then be used for settling ties between identical blocks submitted by different NEMOs. More information are available in paragraph 5.4.4.

5.4.1. Linked Block Orders

Block orders can be linked together, i.e. the acceptance of individual block orders can be made dependent on the acceptance of other block orders. The block which acceptance depends on the acceptance of another block is called "child block", whereas the block which conditions the acceptance of other blocks is called "parent block".

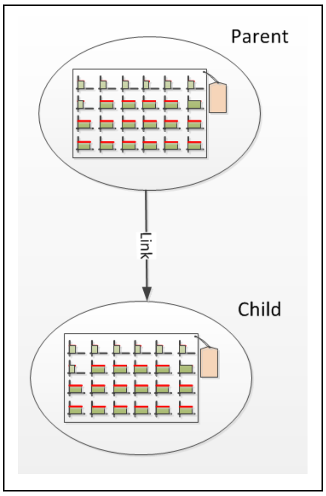


Figure 12 – Linked block orders

The rules for the acceptance of linked block orders are the following:

1. The acceptance ratio of a parent block is greater than or equal to the highest acceptance ratio of its child blocks (acceptance ratio of a child block can be at most the lowest acceptance ratio among own parent blocks)
2. (Possibly partial) acceptance of child blocks can allow the acceptance of the parent block when:
    a. the surplus of a family is non-negative
    b. leaf blocks (block order without child blocks) do not generate economic surplus loss
3. A parent block which is *out-of-the-money* can be accepted in case its accepted child blocks provide sufficient surplus to at least compensate the loss of the parent.
4. A child block which is *out-of-the-money* cannot be accepted even if its accepted parent provides sufficient surplus to compensate the loss of the child, unless the child block is in turn parent of other blocks (in which case rule 3 applies).

In an easy common configuration of two linked blocks, the rules are easy. The parent can be accepted alone, but not the child that always needs the acceptance of the parent first. The child can "save" the parent with its surplus, but not the opposite.

5.4.2. Block Orders in an Exclusive group

An Exclusive group is a set of block orders for which the sum of the accepted ratios cannot exceed 1. In the particular case of blocks that have a minimum acceptance ratio of 1 it means that at most one of the blocks of the exclusive group can be accepted.

Between the different valid combinations of accepted blocks the algorithm chooses the one which maximizes the optimization criterion (*economic surplus*, see Section 7.4).

5.4.3. Flexible Orders

A flexible order is a block order with a fixed price limit, a fixed volume, minimum acceptance ratio of 1, with duration of 1 period. The period is not defined by the participant but will be determined by the algorithm (hence the name "flexible"). The period in which the flexible order is accepted, is calculated by the algorithm and determined by the optimization criterion (see Section 7.4)

5.4.4. Block order tie rule

EUPHEMIA implements block order tie rules to arbitrate between identical blocks, when only some, but not all can be accepted.

Two blocks are considered equal, if they:

- Belong to the same bidding zone;

- Have the same minimum acceptance ratio;
- Have the same price;
- Both are on supply side, or both are on demand side;
- Are defined on the same periods and are offering the same quantities on each period
- Belong to the same exclusive group
- Have no links

For this case economic criteria are insufficient to arbitrate: accepting on or the other will result in identical economic surplus. Instead some secondary criteria are used to make the arbitration, and allow ties to be deterministically broken:

1. A block with an earlier last modification timestamp will be prioritized;

With the introduction of the MNA there is also the need to arbitrate between identical blocks, which were submitted by different NTHs. The initial criterion of the time stamps has been maintained.

On other hand, the second criterion cannot be applied anymore, as ids from the local trading systems are not coordinated. E.g. if NTHs 1 and 2 use a continuous sequence of increasing ids to identify their blocks, but NTH 1 is higher up in its sequence than NTH 2, the NTH 2 blocks will always be prioritized, and the NTHs will not be treated equally.

To avoid unequal treatment the preferred block is selected "randomly": random in the sense bias are avoided, and blocks from one NTH will not be more or less likely to be accepted than blocks from another NTH.

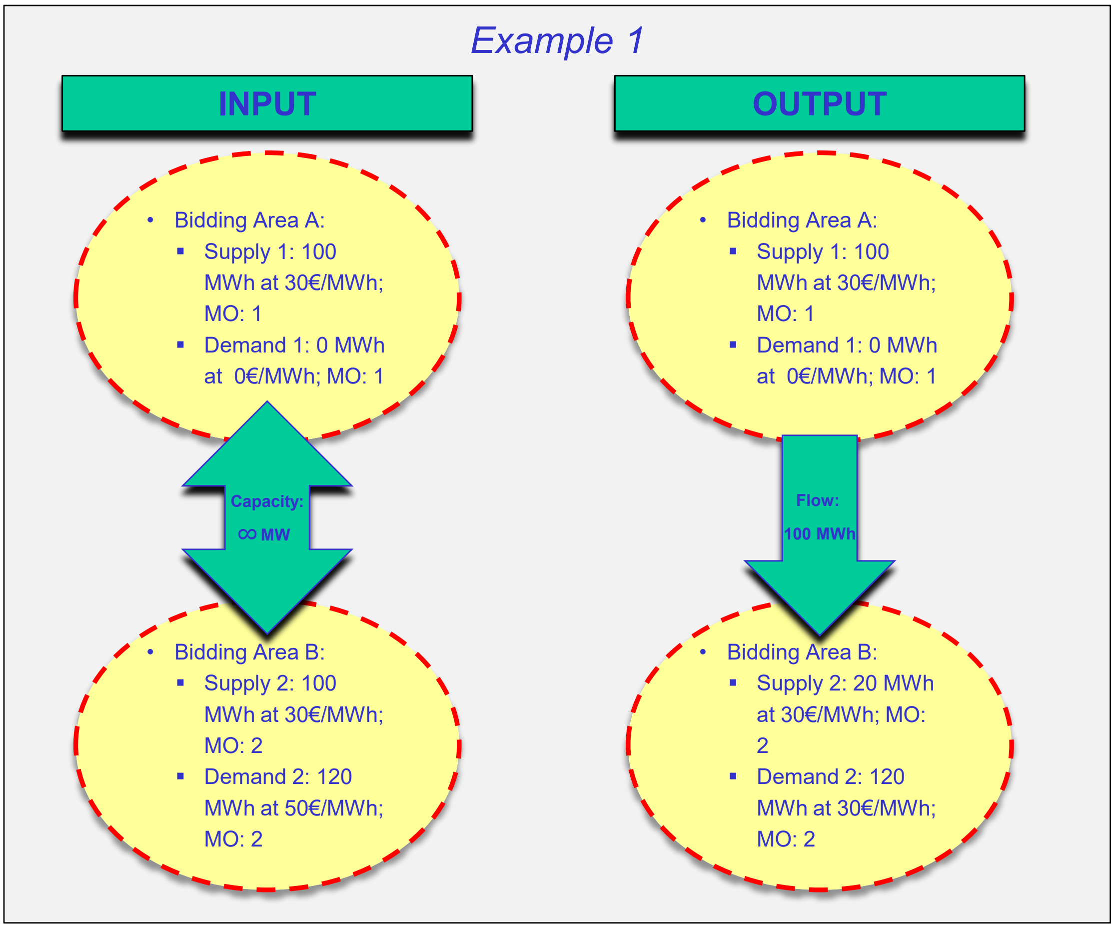
In order to be sure EUPHEMIA behaviour is repeatable, repeatable randomness is applied. This is managed by using the hashes that were compiled for each block (on the basis of the different parameters describing the blocks). These hashes will be used to settle ties, and should be sufficiently random to meet this fairness objective.

5.5. Merit Orders and PUN Orders

5.5.1. Merit Orders

Merit orders are individual step orders defined at a given period for which is associated a so-called merit order number.

A merit order number is unique per period and order type (Demand; Supply; PUN) and is used for ranking merit orders in the *bidding zones* containing this order type. The lower the merit order number, the higher the priority for acceptance. More precisely, when, within an uncongested set of adjacent *bidding zones*, several merit orders have a price that is equal to the market clearing price, the merit order with the lowest merit order number should be accepted first unless constrained by other network conditions.


Example 1
INPUT
OUTPUT
Bidding Area A:
Supply 1: 100 MWh at 30€/MWh; MO: 1
Demand 1: 0 MWh at 0€/MWh; MO: 1
Capacity: ∞ MW
Bidding Area B:
Supply 2: 100 MWh at 30€/MWh; MO: 2
Demand 2: 120 MWh at 50€/MWh; MO: 2
Bidding Area A:
Supply 1: 100 MWh at 30€/MWh; MO: 1
Demand 1: 0 MWh at 0€/MWh; MO: 1
Flow: 100 MWh
Bidding Area B:
Supply 2: 20 MWh at 30€/MWh; MO: 2
Demand 2: 120 MWh at 30€/MWh; MO: 2

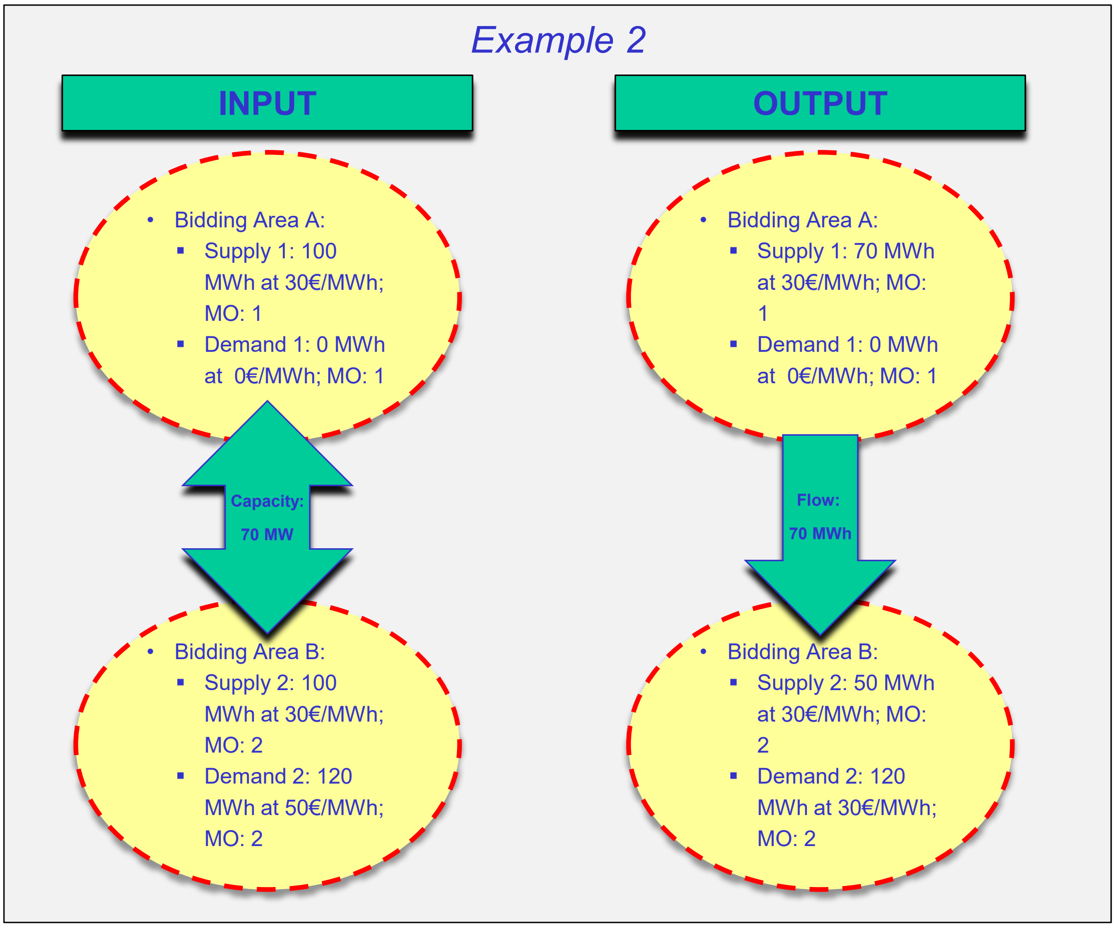

Example 2
INPUT
OUTPUT
Bidding Area A:
Supply 1: 100 MWh at 30€/MWh; MO: 1
Demand 1: 0 MWh at 0€/MWh; MO: 1
Capacity: 70 MW
Bidding Area B:
Supply 2: 100 MWh at 30€/MWh; MO: 2
Demand 2: 120 MWh at 50€/MWh; MO: 2
Bidding Area A:
Supply 1: 70 MWh at 30€/MWh; MO: 1
Demand 1: 0 MWh at 0€/MWh; MO: 1
Flow: 70 MWh
Bidding Area B:
Supply 2: 50 MWh at 30€/MWh; MO: 2
Demand 2: 120 MWh at 30€/MWh; MO: 2

Figure 13: Merit Orders examples

5.5.2. PUN Orders

PUN orders are a particular type of demand merit orders. They differ from classical demand merit orders in such sense that they are cleared at the *PUN price* (PUN stands for "Prezzo Unico Nazionale") rather than the *bidding zone* market clearing price (i.e. a PUN order with an offered price lower than market clearing price of its associated *bidding zone*, but higher than *PUN price* would be fully accepted by EUPHEMIA).

For each period, the values of the accepted PUN merit orders volumes multiplied by the *PUN price* is equal to the value of the accepted PUN merit orders volumes multiplied by the corresponding *market clearing price*s (up to a defined tolerance named PUN imbalance[2]), according to the following Formula:

$$P_{PUN} \times \Sigma_z Q_z = \Sigma_z P_z \times Q_z \pm \Delta$$

With:

- P_{PUN}: *PUN price*
- Q_z: Volumes consumed in *bidding zone* z
- P_z: Price of *bidding zone* z
- Δ: PUN imbalance

In case of more than one PUN order submitted at a price equal to *PUN price*, the merit order number rule is applied to PUN orders as well. PUN orders will no longer be an input of EUPHEMIA starting from January 1, 2025.

6. Cross product matching between different MTUs

With the introduction of support for 15′ time resolution, maintaining support for 60′ (and also 30′) was still deemed desirable. Therefore all three time resolutions are supported by Euphemia, either different time resolutions between different bidding zones, or multiple time resolutions within a single bidding zone.

Products of different time resolutions can "cross match": e.g. 1MW of 60′ demand can be met by 1MW of 15′ supply in each of the quarters underlying the 60′ demand order. This cross matching can happen within the same bidding zone, but also with orders in adjacent bidding zones. This latter case introduces a dependency on the time resolution of the border. Example:

- Imagine a 60′ bidding zone connected to a 15′ bidding zone through a 60′ line. The 60′ orders can be matched against the 15′ orders, and a 60′ flow (or scheduled exchange) supports this;

[2] In other words, the value (PUN Volume * *PUN price*) must be able to refund producers (who receives the price of their bidding zone), congestion rents and a PUN imbalance.

- Imagine 2 15′ bidding zones connected by a 60′ line. Rather than directly matching the 15′ orders in one bidding zone with those of the other 15′ bidding zone, they instead can cross match against the 60′ line: 4x15′ orders in one bidding zone can match with corresponding 4x15′ orders in the other bidding zone, resulting in a 60′ flow (or scheduled exchange) to support this.

The cross match logic applies to most supported order types:
- Aggregated period orders (piecewise linear curves / step curves / hybrid curves);
- Complex orders;
- Scalable complex orders;
- Block orders (including linked block orders / exclusive groups / flexible order);
- Merit orders;

Note: the only unsupported product is the PUN merit order, which is not compatible with 15′ MTU.

Since the same 60′ product can now be matched by either a 60′ order, 2x30′ orders or 4x15′ orders, we potentially create arbitrage opportunities between the different products. To prevent this, Euphemia imposes an "average rule":
The 30′ clearing price equals the average of the underlying 15′ clearing prices;
The 60′ clearing price equals the average of the underlying 30′ clearing prices.

Since this rule exists only 15′ clearing prices will be provided as an official output from EUPHEMIA (or only prices that correspond to the MTU of the bidding zone), and the prices for the coarser time resolution follow from this definition.

Due to this relation between the prices of different time resolutions, there exist corner cases where (marginal) orders for coarser time resolutions may induce the clearing prices of the finer time resolutions to go above the maximum clearing price, or below the minimum clearing price. If we would only clip the prices to be within bounds after this happens, we would break the average rule. This topic is explore in more detail in Annex C – missing and extra money management.

The rules that govern the acceptance of period orders are:

1. One demand (respectively, supply) order is 'in-the-money' when the price of the order is higher (respectively, lower) than the value of the market clearing price(s). Any 'in-the-money' MTU order must be fully accepted. Any order which covers more than one MTU may be paradoxically rejected.
2. One demand (respectively, supply) order is 'out-of-the-money' when the price of the order is lower (respectively, higher) than the value of

the market clearing price(s). Any out-of-the-money order must be rejected.

3. One demand or supply order is 'at-the-money' when the price of the order is equal to the value of the market clearing price(s). Any 'at-the-money' order can be either accepted (fully or partially) or rejected.

7. EUPHEMIA Algorithm

7.1. Preamble: order aggregation

In the following sections, EUPHEMIA solving process is presented.

However, it is important to notice that **EUPHEMIA core computation is performed at bidding zonal level**. Indeed, as presented earlier in the document (4.5.1 and 4.6.1), orders are defined at NTH level but all orders within a same bidding zone are subjected to an identical market clearing price (due to the absence of limitation in terms of flows either between SAs or between NTHs).

While block orders and complex orders remain individually defined, all curve orders from the different NTHs of each bidding zone will be aggregated by EUPHEMIA into a single set of curves for each period for each time resolution that is coarser or equally coarse as the MTU of the bidding zone, as a pre-processing step. Aggregating orders at a bidding zone level allows simplifying EUPHEMIA mathematical model: this way, SA and NTH topologies need not be considered, preventing significant degradation of the algorithm performance.

The type of the aggregated curve will depend on that of the underlying NTH curve types: if all NTHs are all either stepwise or piecewise curves, the generated aggregated curve shall result (respectively) into stepwise and piecewise curves. If NTH curves are however both stepwise and piecewise curves, the resulting curves shall have a hybrid type.

A curve can be considered a function that associates a quantity of Power offered for a time resolution (thus energy if you multiply the power with the duration of the time resolution) with a price (in Euros/MWh):

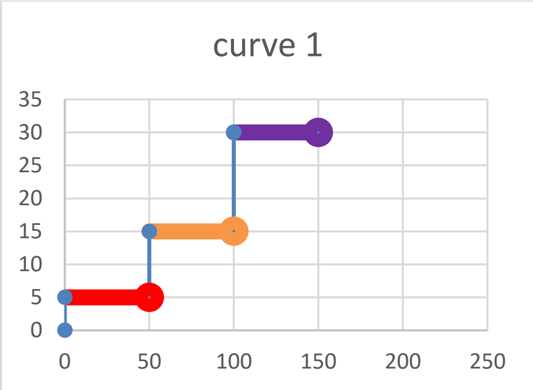
$C(Q) = P$, is a curve (demand or supply) that makes this association.
We can also consider the inverse:
$C^{-1}(P)=Q$

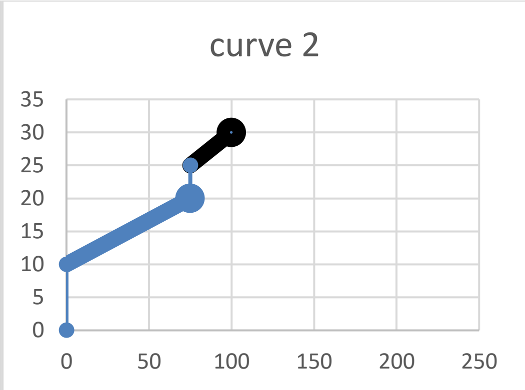
To aggregate 2 (or more) curves we add the inverse functions over the full domain of this inverse function (i.e. between the minimum and maximum price):
$C_{1+2}^{-1}(P)= C_1^{-1}(P)+ C_2^{-1}(P)$
The resulting aggregated curve is to simply re-invert this function.

Mind: the curves are not quite functions, since we allow steps in STEPWISE and HYBRID curves. We can still aggregate by adding these steps and allowing steps in our aggregated curve "function".

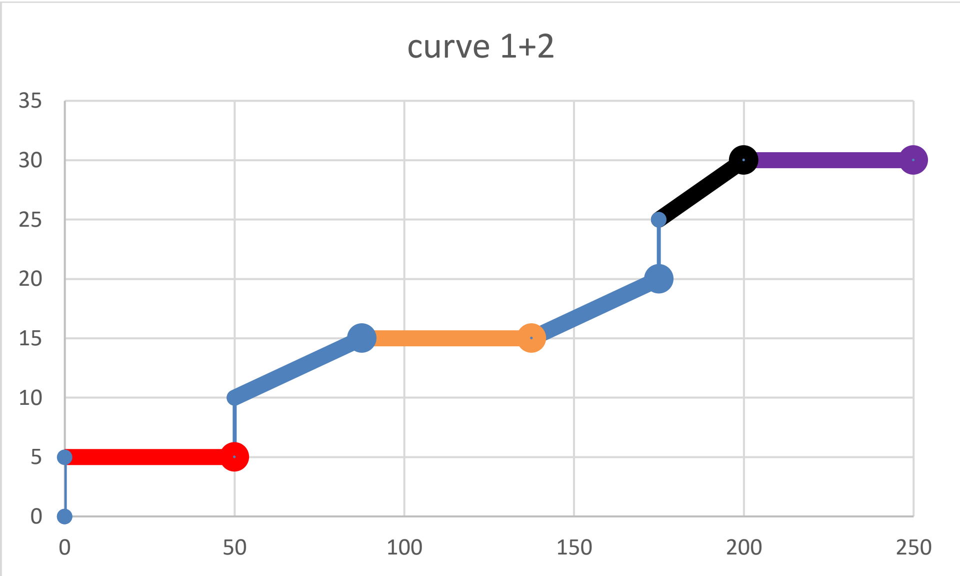
Example
Consider below (supply) curves 1 and 2. Prices (EUR/MWh) on the vertical axis, quantities (MW) on the horizontal axis.

Curve 1 is a step curve with 3 steps, each 50MW at prices 5, 15 and 30 euros respectively;
Curve 2 is a piecewise linear curve, with a 75MW slope from 10 to 20 euros, and a 25MW slope from 25 to 30 euros.





We aggregate them by adding at each price level the corresponding quantities to get:



The colours allow you to identify where the segments came from. Note that the blue segment from curve 2 is split into two in the aggregated curve, as the 15 euro step from curve 1 had its price precisely in the middle of the blue segment.

To retrieve the results at NTH level, EUPHEMIA also implements a disaggregation post-processing step, once solutions have been found.

7.2. Cluster order aggregation

Similar to the aggregation at the bidding zone level from the previous section, Euphemia can also aggregate curves for groups (or clusters) of bidding zones. A functionality is available to configure clusters of bidding zones for which the curves can be aggregated. This information can be used for the publication of anonymised aggregated curves per each period for each time resolution that is coarser or equal to the time resolution of the cluster

For more information where to find the publications of the aggregated curves, please consult:

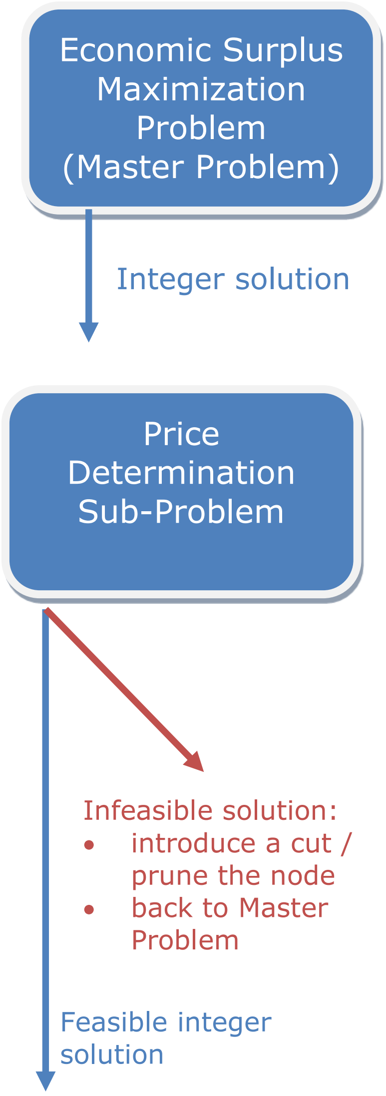
https://www.nemo-committee.eu/aggregated_curves

7.3. Overview

As mentioned previously, EUPHEMIA is the algorithm that has been developed to solve the Day-Ahead European Market Coupling problem. EUPHEMIA matches energy demand and supply for all the periods of a single day at once while taking into account the market and network constraints. The main objective of EUPHEMIA is to maximize the *economic surplus*, *i.e.* the total market value of the Day-Ahead auction expressed as a function of the *consumer surplus*, the *supplier surplus*, and the *congestion rent* including tariff rates on interconnectors if they are present. EUPHEMIA returns the *market clearing prices*, the matched volumes, and the *net position* of each *bidding zone* as well as the flow through the interconnectors. It also returns the selection of block, complex, merit, and PUN orders that will be executed. For curtailable blocks the selection status will indicate the accepted percentage for each block.

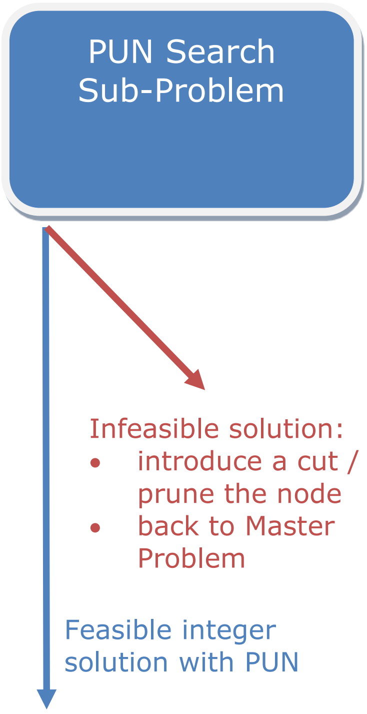
By ignoring the particular requirements of the block, complex, merit and PUN orders, the market coupling problem resolves into a much simpler problem which can be modelled as a Quadratic Program (QP) and solved using commercial off-the-shelf solvers. However, the presence of these orders renders the problem more complex. Indeed, the "kill-or-fill" property of block orders and the minimum income condition (MIC) of complex orders requires the introduction of binary (i.e. 0/1) variables. Moreover, the strict consecutiveness requirement of merit and PUN orders adds up to the complexity of the problem.

In order to solve this problem, EUPHEMIA runs a combinatorial optimization process based on the modelling of the market coupling problem. The reader can refer to the Annex C for a more detailed mathematical formulation of the problem. EUPHEMIA aims to solve a economic surplus maximization problem (also referred to as the master problem) and three interdependent sub-problems, namely the price determination sub-problem, the PUN search sub-problem and the volume indeterminacy sub-problem.

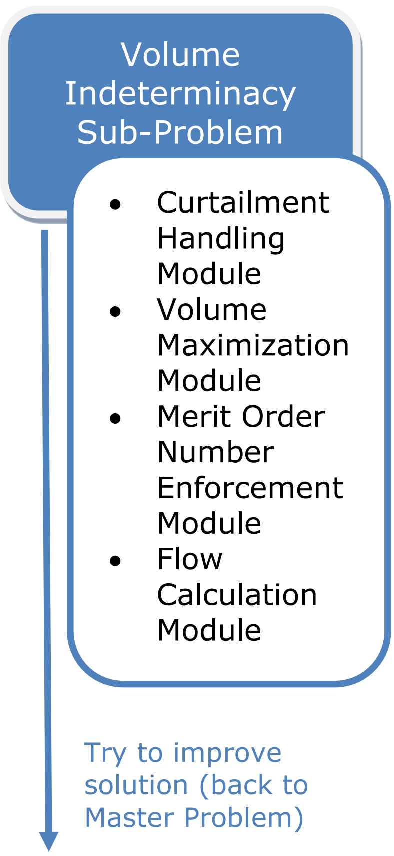


In the economic surplus maximization problem, EUPHEMIA searches among the set of solutions (solution space) for a good selection of block and MIC orders that maximizes the *economic surplus*. In this problem, the PUN and merit orders requirements are not enforced. Once an integer solution has been found for this problem, EUPHEMIA moves on to determine the *market clearing prices*.

The objective of the price determination sub-problem is to determine, for each *bidding zone*, the appropriate *market clearing price* while ensuring that no block and complex MIC orders are *paradoxically accepted* and that the flows price-network requirements are respected (more precisely: that the primal-dual relations are satisfied, cf. Annex C). If a feasible solution could be found for the price determination sub-problem, EUPHEMIA proceeds with the PUN search sub-problem. However, if the sub-problem does not have any solution, we can conclude that the block and complex orders selection is not acceptable, and the integer solution to the economic surplus maximization problem must be rejected. This is achieved by adding a cut to the economic surplus maximization problem that renders its current solution infeasible. Subsequently, EUPHEMIA resumes the economic surplus maximization problem searching for a new integer solution for the problem.



The objective of the PUN search sub-problem is to find valid PUN volumes and prices for each period of the day while satisfying the PUN imbalance constraint and enforcing the strong consecutiveness of accepted PUN orders. When the PUN search sub-problem is completed, EUPHEMIA verifies that the obtained PUN solution does not introduce any *paradoxically accepted block/complex orders*. If some orders become *paradoxically accepted*, a new cut is introduced to the economic surplus maximization problem that renders the current solution infeasible. Otherwise, EUPHEMIA proceeds with the lifting of volume indeterminacies.



In the previous sub-problems, the algorithm has determined the *market clearing prices* for each *bidding zone*, the *PUN prices* and volumes for the area with PUN orders, and a selection of block and complex MIC orders that are feasible all together. Though, there might exist several aggregated period volumes, *net positions*, and bidding zone line flows that are coherent with these prices and that yield the same economic surplus. Among all these possible solutions, EUPHEMIA pays special attention to the *price-taking orders*, enforces the merit order number, and maximizes the traded volume.

The flow calculation module here also takes into account both scheduling area and NEMO trading hubs topologies. More details can be found in section 7.9.5.

7.4. *Economic surplus* Maximization Problem (Master Problem)

As mentioned previously, the objective of this problem is to maximize the *economic surplus*, *i.e.* the total market value of the Day-Ahead auction. The *economic surplus* is computed as the sum of the *consumer surplus*, the *supplier surplus*, and the *congestion rent*. The latter takes into account the presence of tariff rates for the flows through defined interconnectors.

In case there is the risk of a curtailment situation in an area where Flow Based constraints apply, a special penalty is applied in the objective function for the non-acceptance of price taking demand. This is linked to the curtailment sharing rules, which are described in 7.9.2

EUPHEMIA ensures that the returned results are coherent with the following constraints (see Chapters 4 and 5):

- The acceptance criteria for aggregated period demand and supply curves and merit orders
- The fill-or-kill requirement of block orders
- The scheduled stop, load gradient, and minimum income condition of complex orders and scalable complex orders
- The capacities and ramping constraints imposed on the ATC interconnectors while taking into account the losses and the tariff rates if applicable.
- The flow limitation through some critical elements of the network for *bidding zones* managed by the flow-based network model. All bidding zones should be balanced: the net position equals the total export minus the total imports for this zone, and this should match the zone's imbalance: the difference between total matched supply and total matched demand.
- The net position ramping should be respected;

It should be noted that the strict consecutiveness requirement of merit and PUN orders is not enforced in this problem. In other words, the merit orders are considered in this problem as aggregated period orders while, the PUN orders are just ignored. The main difficulty of the *economic surplus* maximization problem resides in selecting the block/MIC orders that are to be accepted and those to be rejected. The particularity of the block and MIC orders lies in the fact that they require the introduction of 0/1 variables in order to model their acceptance (0: rejected order, 1: accepted order). The discrete nature of these decision variables is referred to as the integrality constraint. The solution of this problem requires some decision variables to be integer (0/1) and the overall problem can be modelled as a Mixed-Integer Quadratic Program (MIQP).

A possible approach to solve such an MIQP problem is to use the branch-and-cut method. The branch-and-cut method is a very efficient technique for solving a wide variety of integer programming problems. It involves running a branch-and-bound algorithm and using cutting planes to tighten the QP relaxations. In the sequel, we will describe how the branch-and-cut method can be adapted to our particular *economic surplus* maximization problem and how cutting planes will be generated in the subsequent sub-problems in order to reduce the number and range of solutions to investigate.

7.4.1. Overview

EUPHEMIA starts by solving the initial MIQP problem where none of the variables is restricted to be integer. The resulting problem is called the integer relaxation of the original MIQP problem. For instance, relaxing the fill-or-kill constraint, *i.e.* the integrality constraint on the acceptance of the block orders, is equivalent to allowing all the block orders to be partially executed.

Because the integer relaxation is less constrained than the original problem, but still aims at maximizing *economic surplus*, it always gives an upper bound on attainable *economic surplus*. Moreover, it may happen that the solution of the relaxed problem satisfies all the integrality constraints even though these constraints were not explicitly imposed. The obtained result is thus feasible with respect to the initial problem and we can stop our computation: we got the best feasible solution of our MIQP problem. Note that this is rarely the case and the solution of the integer relaxation contains very often many fractional numbers assigned to variables that should be integer values.

7.4.2. Branching

In order to move towards a solution where all the constraints, including the integrality constraints, are met, EUPHEMIA will pick a variable that is violating its integrality constraint in the relaxed problem and will construct two new instances as following:

- The first instance is identical to the relaxed problem where the selected variable is forced to be smaller than the integer part of its current fractional value. In the case of 0/1 variables, the selected variable will be set to 0. This will correspond, for instance, to the case where the block order will be rejected in the final coupling solution.
- The second instance is identical to the relaxed problem where the selected variable is forced to be larger than the integer part of its current fractional value. In the case of 0/1 variables, the selected variable will be set to 1. This will correspond, for instance, to the case where the block order will be accepted in the final coupling solution.

Duplicating the initial problem into two new (more restricted) instances is referred to as branching. Exploring the solution space using the branching method will result in a tree structure where the created problem instances are referred to as the nodes of the tree. For each created node, the algorithm tries to solve the relaxed problem and branches again on other variables if necessary. It should be highlighted that by solving the relaxed problem at each of the nodes of the tree and taking the best result, we have also solved the initial problem (i.e. the problem in which none of the variables is restricted to be integer).

7.4.3. Fathoming

Expanding the search tree all the way till the end is termed as fathoming. During the fathoming operation, it is possible to identify some nodes that do not need to be investigated further. These nodes are either pruned or terminated in the tree which will considerably reduce the number of instances to be investigated. For instance, when solving the relaxed problem at a certain node of the search tree, it may happen that the solution at the current node satisfies all the integrality restrictions of the original MIQP problem. We can thus conclude that we have found an integer solution that still needs to be proved feasible. This can be achieved by verifying that there exist valid *market clearing prices* for each *bidding zone* that are coherent with the market constraints. For this purpose, EUPHEMIA moves on to the price determination sub-problem (see section 7.5). If the latter sub-problem finds a valid solution for the current set of blocks/complex orders, we can conclude that the integer solution just found is feasible. Consequently, it is not required to branch anymore on this node as the subsequent nodes will not provide higher *economic surplus*. Otherwise, if no valid solution could be found for the price determination sub-problem, we can conclude that the current block and complex order selection is unacceptable. Thus, a new instance of the *economic surplus* maximization problem is created where additional constraints are added to the *economic surplus* maximization problem that renders the previous integer solution infeasible (see section 7.4.4).

Let us denote the best feasible integer solution found at any point in the search as the incumbent. At the start of the search, we have no incumbent. If the integer feasible solution that we have just found has a better objective function value than the current incumbent (or if we have no incumbent), then we record this solution as the new incumbent, along with its objective function value. Otherwise, no incumbent update is necessary and we simply prune the node.

Alternatively, it may happen that the branch, that we just added and led to the current node, has added a restriction that made the QP relaxation infeasible. Obviously, if this node contains no feasible solution to the QP relaxation, then it contains no integer feasible solution for the original MIQP problem. Thus, it is not necessary to further branch on this node and the current node can be pruned.

Similarly, once we have found an incumbent, the objective value of this incumbent is a valid lower bound on the *economic surplus* of our *economic surplus* maximization problem. In other words, we do not have to accept any integer solution that will yield a solution of a lower *economic surplus*. Consequently, if the solution of the relaxed problem at a given node of the search tree has a smaller *economic surplus* than that of the incumbent, it is not necessary to further branch on this node and the current node can be pruned.

7.4.4. Cutting

Introducing cutting planes is the other most important contributor of a branch-and-cut algorithm. The basic idea of cutting planes (also known as "cuts") is to progressively tighten the formulation by removing undesirable solutions. Unlike the branching method, introducing cutting planes creates a single new instance of the problem. Furthermore, adding such constraints (cuts) judiciously can have an important beneficial effect on the solution process.

As just stated, whenever EUPHEMIA finds a new integer solution with a better *economic surplus* than the incumbent solution, it moves on to the price determination sub-problem and subsequent sub-problems. If in these sub-problems, we find out that the sub-problem is infeasible, we can conclude that the current block and complex order selection is unacceptable. Thus, the integer solution of the *economic surplus* maximization problem must be rejected. To do so, specific local cuts are added to the *economic surplus* maximization problem that renders the current selection of block and complex orders infeasible. Different types of cutting planes can be introduced according to the violated requirement that should be enforced in the final solution. For instance, if at the end of the price determination sub-problem, a block order is *paradoxically accepted*, the proposed cutting plane will force some block orders to be rejected so that the prices will change and will eventually make the block order no longer *paradoxically accepted*. Further types of cutting planes will be introduced in the subsequent sub-problems.

7.5. Price Determination Sub-problem

In the master problem, EUPHEMIA has determined an integer solution with a given selection of block and complex orders. In addition, EUPHEMIA has also determined the matched volume of merit and aggregated period orders. In this sub-problem, EUPHEMIA must check whether there exist *market clearing prices* that are coherent with this solution while still satisfying the market requirements. More precisely, EUPHEMIA must ensure that the returned results satisfy the following constraints:

- The *market clearing price* of a given *bidding zone* at a specific period of the day is coherent with the offered prices of the demand orders and the desired prices of the supply orders in this particular market.
- The *market clearing price* of a *bidding zone* is compatible with the minimum and maximum price bounds fixed for this particular market.

However, the solution of this price determination sub-problem is not straightforward because of the constraints preventing the *paradoxical acceptance of block and MIC orders*, or preventing the presence of *non-intuitive FB results*. Indeed, whenever EUPHEMIA deems that the price determination sub-problem is infeasible, it will investigate the cause of infeasibility and a specific type of cutting plane will be added to the *economic surplus* maximization problem aiming at enforcing compliance with the corresponding requirement. This cutting plane will discard the current selection of block and complex orders.

- In order to prevent the *paradoxical acceptance of block orders*, the introduced cutting plane will reject some block orders that are *in-the-money*. Special attention will be paid when generating these cuts in order to prevent rejecting *deep-in-the money* orders.
- In order to prevent the acceptance of complex orders that do not satisfy their minimum income condition, the introduced cutting plane

will reject the complex orders that will most likely not fulfil their minimum income condition.

- When the market coupling problem at hand features both block and complex orders, EUPHEMIA associates both cutting strategies in a combined cutting plane.

Cuts will also be generated under the following circumstances:

- Furthermore, if the bilateral intuitiveness mode is selected for the flow-based model, the prices obtained at the end of the price determination sub-problem must satisfy an additional requirement. This requirement states that there cannot be *adverse flows*, *i.e.* flows exporting out of more expensive markets to cheaper ones. If the intuitiveness property is not satisfied, appropriate cutting planes are added as well to the *economic surplus* maximization problem.
- In the presence of losses in a situation where a market clears at a negative price bi-directional flows may occur: energy is send back and forth between two areas only to pick up losses.

  Algorithmically this makes sense: when a market clears at a negative price, it is willing to pay for destroying energy (e.g. through losses). However physically it is nonsensical: energy can only be scheduled in one direction. To avoid this situation EUPHEMIA will generate a cut forcing one or the other flow to be zero.

At this stage, we have obtained a feasible integer selection of block and complex orders along with coherent *market clearing prices* for all markets. Next, EUPHEMIA moves on to the PUN search sub-problem where it enforces the strong consecutiveness of the merit and PUN orders as well as the compliance with the PUN imbalance constraint.

Partial decoupling cases

To support partial decoupling cases in the Core region, no order data may be present in one of the bidding zones, and it will be disconnected from the rest of the topology. I.e. there is no proper basis to set a meaningful price for such partially decoupled bidding zone. Core TSOs requested to set the price of the decoupled Core bidding zones to the average price of adjacent (non-decoupled) bidding zones.

If those adjacent bidding zones have a finer time resolution than that of the decoupled bidding zone, the average rule is used. If they have coarser time resolution, then the price of the corresponding parent period is used.

7.5.1. Branch-and-Cut Example

Here is a small example of the execution of the Branch-and-Cut algorithm (Figure 14).

At the start of the algorithm, we do not have an incumbent solution. EUPHEMIA first solves the relaxed *economic surplus* maximization problem where all the integrality constraints have been relaxed (Instance A). Let us assume that the solution of this problem has a *economic surplus* equal to 3500 but has two fractional decision variables related to the acceptance of the block orders ID_23 and ID_54. At this stage, we can conclude that the upper bound on the attainable *economic surplus* is equal to 3500.

Next, EUPHEMIA will pick a variable that is violating its integrality constraint (block order ID_23, for instance) and will branch on this variable. Thus, two new instances are constructed: Instance B where the block order ID_23 is rejected (associated variable set to 0) and Instance C where the block order ID_23 is accepted (associated variable set to 1). Then, EUPHEMIA will select one node that is not yet investigated and will solve the relaxed problem at that node. For example, let us assume that EUPHEMIA selects Instance B to solve and finds a solution where all the variables associated with the acceptance of block and complex orders are integral with a *social economic surplus* equal to 3050. Furthermore, we assume that the price determination sub-problem was successful and that a valid solution could be obtained. We can conclude that the solution of Instance B is thus feasible and can be marked as the incumbent solution of the problem. In addition, the obtained *economic surplus* is a lower bound on any achievable *economic surplus* and it is not necessary to further branch on this node.

EUPHEMIA continues exploring the solution space and selects Instance C to solve. Let us assume that an integer solution was found with a *economic surplus* equal to 3440. As the obtained *economic surplus* is higher than that of the incumbent, EUPHEMIA moves on to the price determination sub-problem but let us assume that no valid *market clearing prices* could be found for this sub-problem. In this case, a local cut will be introduced to the *economic surplus* maximization problem. More precisely, an instance D is created identical to instance C where an additional constraint is added to render the current selection of block and complex orders infeasible. At this stage, we can conclude that the upper bound on the attainable *economic surplus* is equal to 3440.

Now, let us assume that when solving the instance D of the problem, we get a solution with a *economic surplus* equal to 3300 and a fractional decision variable related to the acceptance of the block order ID_30. As carried out previously, we need to branch on this variable. Thus, two new instances are constructed: Instance E where the block order ID_30 is rejected (associated variable set to 0) and Instance F where the block order ID_30 is accepted (associated variable set to 1). After solving the relaxed problem of Instance E, we assume that the obtained solution is integer with a *economic surplus* equal to 3200. This *economic surplus* is higher than that of the incumbent, so we try to solve the price determination sub-problem. We assume that the price determination sub-problem has a valid solution. Thus, the current solution for Instance E is feasible and is set as the new incumbent solution. We note that the lower bound on any achievable *economic surplus* is now equal to 3200.

Similarly, after solving the relaxed problem of Instance F, we assume that the obtained solution has a *economic surplus* equal to 3100 along with some fractional decision variables. As this solution has a lower *economic surplus* than that of the incumbent, there is no need to further branch on this node and the current node can be pruned.

Figure 14 shows the search tree associated with our example.

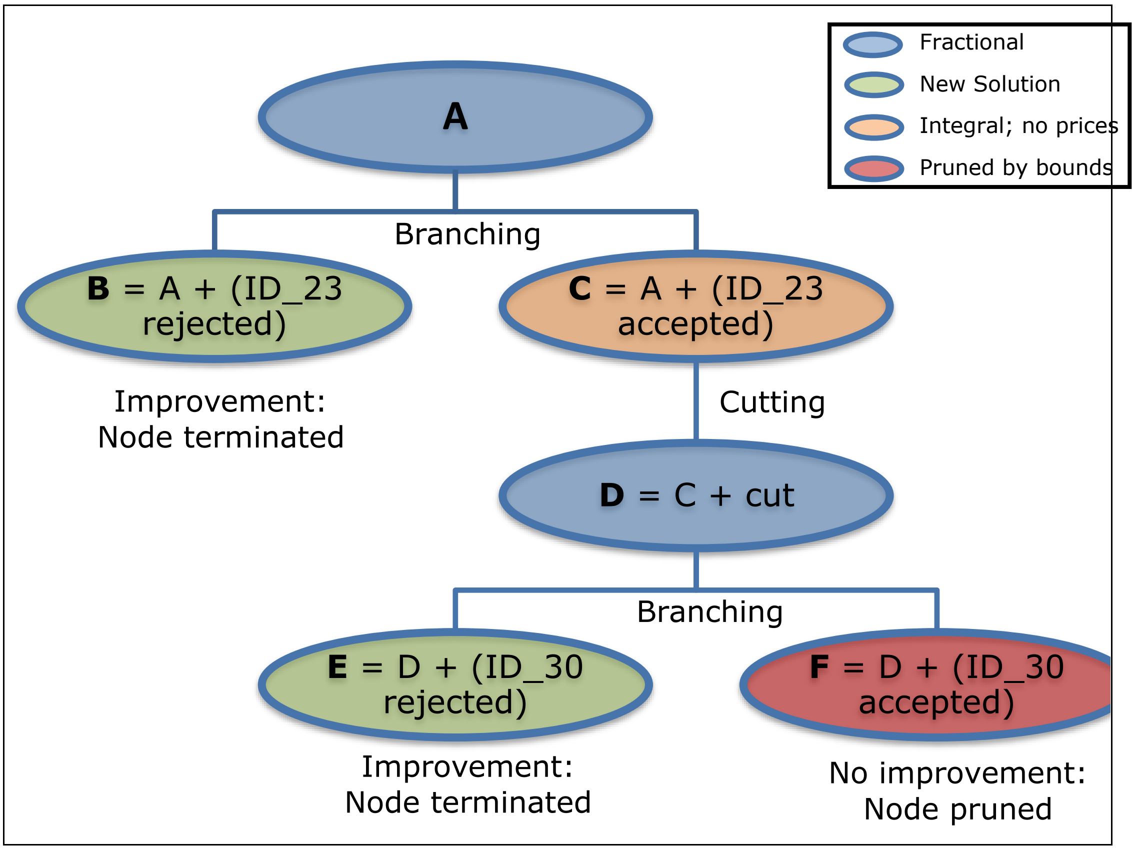


Figure 14 - Branch-and-Cut example

7.6. PUN Search Sub-problem

In order to avoid *paradoxically accepted* PUN orders, PUN (see Section 7.6) cannot be calculated as ex post weighted average of market price, but it must definitely be determined in an iterative process. Consider the following example:

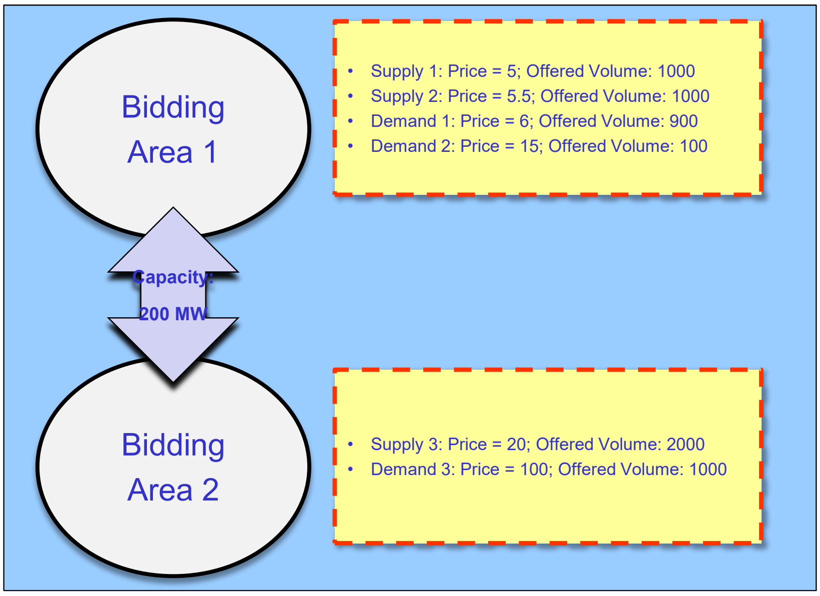


Figure 15 – PUN acceptance

If in Figure 16, Demand 1, Demand 2 and Demand 3 Orders were "simple" demand merit orders, then the market results would be:

- *Bidding zone* 1:
  - *Market clearing price*: 5.5 €/MWh;
  - Executed Supply Volume: 1000 MWh;
  - Executed Demand Volume: 1000 MWh.
- *Bidding zone* 2:
  - *Market clearing price*: 20 €/MWh;
  - Executed Supply Volume: 1000 MWh;
  - Executed Demand Volume: 1000 MWh.

If Demand 1, Demand 2 and Demand 3 Orders were "PUN" demand merit orders, then this solution is not acceptable. In fact, given a PUN imbalance tolerance=0, PUN calculated as weighted average will be:

[(1000 * 5.5) + (1000 * 20)] / 2000 = 12.75 €/MWh.

In this case, order Demand 1 would be *paradoxically accepted*.

Through an iterative process, the final solution will be the following:

- *Market clearing price* of *Bidding zone* 1: 5 €/MWh;
- *Market clearing price* of *Bidding zone* 2: 20 €/MWh;
- *PUN price*: 20 €/MWh;
- Supply order Supply 1: partially accepted (200 MWh);
- Supply order Supply 2: fully rejected;
- Supply order Supply 3: partially accepted (800 MWh)
- Demand orders Demand 1 and Demand 2: fully rejected;
- Demand order Demand 3: fully accepted;
- Flow from *Bidding zone* 1 to *Bidding zone* 2: 200 MWh;
- Imbalance: (1000 * 20) – (1000 * 20)= 0;
- Economic surplus: (1000 * 100) – [(200 * 5 + 800 * 20)] = 83000 €;

The PUN search is launched as soon as a first candidate solution has been found at the end of the price determination sub-problem (activity 1 in Figure 15). This first candidate solution respects all PCR requirements but PUN. The objective of the PUN search is to find, for each period, valid PUN volumes and prices (activity 2 in Figure 15) while satisfying the PUN imbalance constraint and enforcing the strong consecutiveness of accepted PUN orders.

If the solution found for all periods of the day, is compatible with the solution of the master problem (activity 3 in Figure 17), it means that a solution is found after PRMIC reinsertion (see next section) has been performed. Otherwise, the process will resume calculating, for each period, new valid PUN volumes and prices to apply to PUN Merit orders.

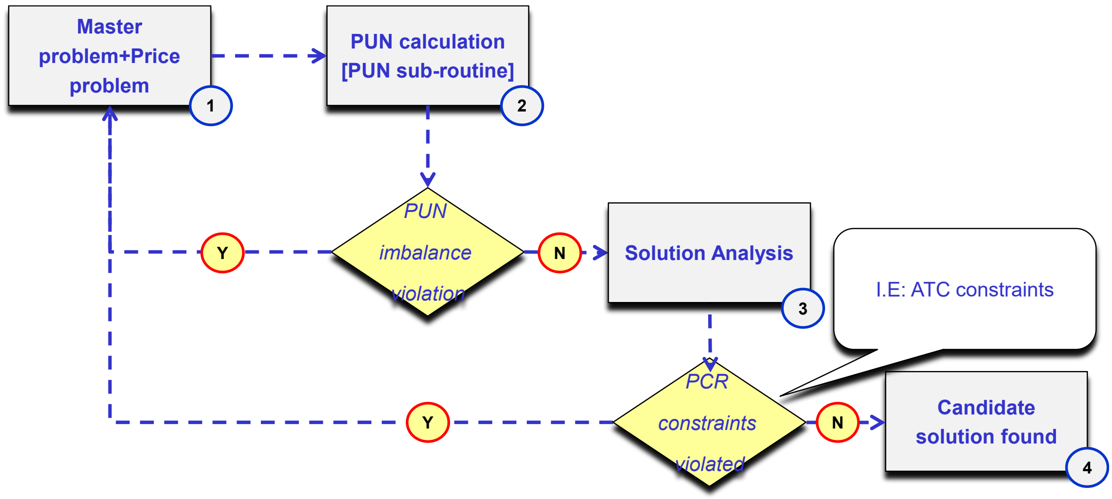


Figure 17 – PUN Search Sub-problem process

The PUN search is essentially a period sub-problem where the requirements are defined on a per period basis, in which:

- Strong consecutiveness of PUN order acceptance is granted: a PUN order at a lower price cannot be satisfied until PUN orders at higher price are fully accepted
- PUN imbalance is within accepted tolerances.

For a given period, the selected strategy consists in selecting the maximum PUN volume (negative imbalance), and then trying to select smaller volumes until a feasible solution is found that minimizes the PUN imbalance.

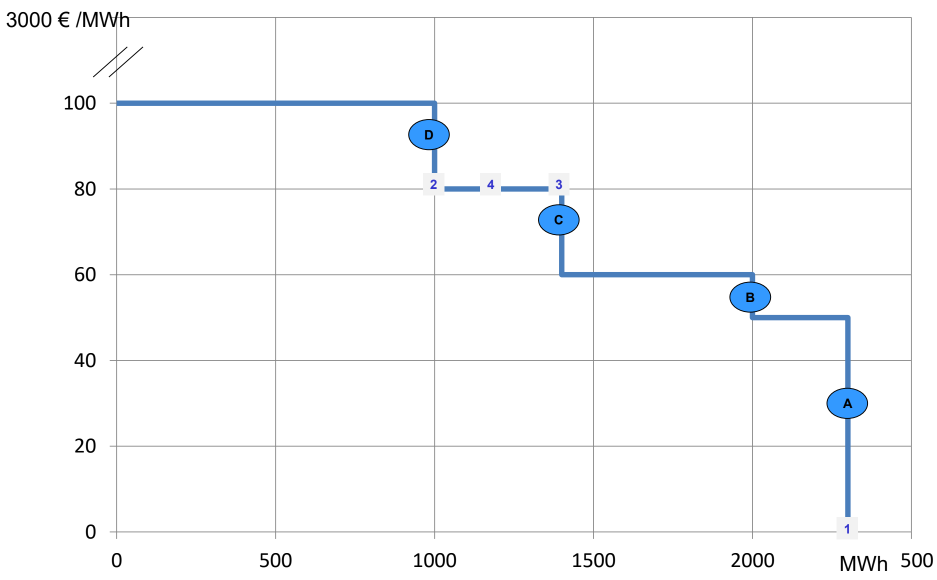


Figure 18 – PUN period curve

EUPHEMIA starts by calculating the PUN imbalance associated with the maximum accepted PUN volume (negative imbalance expected[3]; point 1 in Figure 18). If the PUN imbalance associated with the maximum PUN doesn't violate PUN imbalance tolerance, a candidate solution is found.

[3] PUN consumers paid 0, producers receive market prices. Unless all market prices are equal to 0, imbalance will be negative

On the contrary, EUPHEMIA calculates the price which minimizes PUN imbalance (in Figure 18, analysis on vertical segment A) while the volume is fixed to the maximum accepted PUN volume. If the PUN imbalance calculated in this way is within the PUN imbalance tolerance interval, a candidate solution is found. If not, the next vertical segment (i.e. in Figure 18, vertical segment B), will be analyzed. This process is repeated until between 2 consecutive vertical segments, a change in sign of PUN imbalance is found (i.e. in Figure 18, positive PUN Imbalance in segment D; and negative PUN Imbalance in segment C). In this case, EUPHEMIA fixes the price (i.e. in Figure 18, the horizontal segment between point 2 and 3, to which corresponds a price of 80 €/MWh), and tries to minimize the PUN imbalance, using the volume as decision variable.

If the PUN imbalance calculated in this step is compatible with PUN imbalance tolerance, a candidate solution is found. If not, EUPHEMIA continues the search on the horizontal segment (*i.e.* considering in Figure 18, let point 4 the one associated with PUN imbalance minimization at the price of 80 €/MWh. If in point 4, the imbalance is positive and greater than positive PUN imbalance tolerance, search will be continued in the interval between [4;3]; If in point 4, the imbalance is negative and less than negative PUN imbalance tolerance, the search will be continued in the interval between [2;4]).

PUN SEARCH SUMMARY

1. *Calculation of PUN imbalance associated with maximum accepted PUN volume:*
   - *If minimum PUN imbalance tolerance ≤ calculated imbalance ≤ maximum PUN imbalance: candidate solution found*
   - *If imbalance < minimum PUN imbalance, next vertical segment is analyzed*
2. *Vertical segment analysis: Fixed the volume, minimization of the imbalance*
   - *If minimum PUN imbalance ≤ calculated imbalance ≤ maximum PUN imbalance: candidate solution found*
   - *If imbalance < minimum PUN imbalance, next vertical segment is analyzed*
   - *If imbalance > maximum PUN imbalance, next horizontal segment is analyzed*
3. *Horizontal segments analysis: Fixed the volume, minimization of the imbalance:*
   - *If minimum PUN imbalance ≤ calculated imbalance ≤ maximum PUN imbalance: candidate solution found*
   - *If imbalance < minimum PUN Imbalance, next horizontal segment is analyzed*
   - *If imbalance > maximum PUN Imbalance, next horizontal segment is analyzed*

As soon as PUN search is completed, EUPHEMIA verifies that the obtained PUN solution does not introduce any *paradoxically accepted block orders* or violates any other PCR constraints. If some block orders become *paradoxically accepted* or some other constrains are violated, a new cut is introduced to the economic surplus maximization problem that renders its current solution infeasible. Otherwise, EUPHEMIA proceeds with the PRMIC reinsertion.

7.7. PRMIC reinsertion

Finally, if the PUN sub-problem is successful, the solution returned by EUPHEMIA should be made free of any false paradoxically rejected (scalable) complex MIC/MP order (PRMIC). Thus, once the market clearing prices have been found, EUPHEMIA proceeds with an iterative procedure aiming to verify that all the rejected (scalable) complex MIC/MP orders, that are in-the-money, cannot be accepted in the final solution. For this purpose, EUPHEMIA first determines the list of false PRMIC candidates. Then, EUPHEMIA goes through the list, takes each (scalable) complex MIC/MP order from this list, activates it, and re-executes the price determination sub-problem. Two possible outcomes are expected:

- If the price computation succeeds and the *economic surplus* was not degraded, we can conclude that the PRMIC reinsertion was successful. In this case, a new list of *false PRMIC* candidates is generated and the PRMIC reinsertion module is executed again.
- Conversely, if the price determination sub-problem is infeasible, or the *economic surplus* is reduced, the (scalable) complex MIC/MP order candidate is simply considered as a true PRMIC, and the algorithm picks the next *false PRMIC* candidate. It should be noted that this case will not result to add a new cutting plane to the economic surplus maximization problem.

The PRMIC reinsertion module execution is repeated until no false PRMIC candidate remains. At this stage, we have obtained a feasible integer selection of block and complex orders along with coherent market clearing prices for all markets.

7.8. PRB reinsertion

In much the same way as the PRMIC reinsertion procedure, a module is in charge of reinserting PRBs after a fully valid solution has been found in the Branch-and-Bound tree. This local search approach helps reduce the number of PRBs, and usually leads quickly to a new solution, with a better economic surplus.
As soon as a solution has been stored, a local search algorithm tries to find neighbour solutions where some PRBs are newly activated. The MICs selection is fixed for this step. Of course, just like the PRMICs, not all PRBs may be reactivated. Some of them, when they are reinserted, change the prices in such a way that the solution is not valid anymore. They are true PRBs.
The procedure for the local search stops for each neighbour type when either one of these criteria is met:

- The list of candidate neighbours is empty. In this case, a local search for the next neighbour type is started or the local search stops if all neighbour types were already considered.

- The time limit is getting too close: based on historical performance 3 minutes is required for the remaining sub-problems

After selecting a neighbour solution, it is possible that a new PUN search is needed. The newly activated and deactivated blocks may indeed have invalidated the PUN results, since the imbalance is not enforced by a constraint in this module, contrary to what is done in the PRMIC reinsertion module. In any case, the PRMIC reinsertion procedure and the volume problems are then run to obtain a second fully valid solution.
Like the false PRMIC reinsertion module, this module allows EUPHEMIA to bypass the branch and cut mechanism, by taking a "shortcut" in the tree. The economic surplus of the new solution will be used as a cut-off value to prune other nodes.
Note that the local search module is only applied once at each node where a valid solution is found. After that, the search is resumed in the Branch-and-Bound tree.

A heuristic approach is used at multiple levels in the local search procedure:
We have to restrict the neighbourhood in our search. Thus, we consider only single orders. However, a combination of orders can sometimes lead to better solutions and it can be impossible to reach those solutions via this local search.
The candidate neighbours are given in a certain order. By choosing to reactivate the orders according to this criterion, EUPHEMIA might miss other combinations of activations leading to a solution.
If the price computation fails, no cuts are added. We assume that the reinsertion of the order makes the prices problem infeasible and therefore reject it.

7.9. Volume Indeterminacy Sub-problem

With calculated prices and a selection of accepted block, MIC and PUN orders that provide together a feasible solution to market coupling problem, there still might be several matched volumes, *net positions* and flows coherent with these prices. Among them, EUPHEMIA must select one according to the volume indeterminacy rules, the curtailment rules, the merit order rules and the flow indeterminacy rules. These rules are implemented by solving five closely related optimization problems:

- Curtailment minimization
- Curtailment sharing
  - Partially addressed via the curtailment mitigation in the economic surplus definition;
- Volume maximization
- Merit order indeterminacy
- Flow indeterminacy

7.9.1. Curtailment minimization

A *bidding zone* is said to be in curtailment when the *market clearing price* is at the maximum or the minimum allowed price of that *bidding zone and submitted quantity at these extreme prices if not fully accepted for price-taking orders*. Here we define "*price taking orders*" as buy at max price of sell at min price period orders submitted at the time-resolution of the bidding zone. The curtailment ratio is the proportion of *price-taking orders* which are not accepted. All orders have to be submitted within a (technical) price range set in the respective *bidding zone*. Period supply orders at the minimum price of this range and period demand orders at the maximum price of this range are interpreted as *price-taking orders*, indicating that the member is willing to sell/buy the quantity irrespective of the *market clearing price*.

The first step aims at minimizing the curtailment of these *price-taking* limit orders, *i.e.* minimizing the rejected quantity of *price-taking orders*. More precisely, EUPHEMIA enforces local matching of *price-taking period orders* with period orders from the opposite sense in the same *bidding zone* as a counterpart. Hence, whenever curtailment of *price-taking orders* can be avoided locally on a period basis – *i.e.* the curves cross each other - then it is also avoided in the final results. This can be interpreted as an additional constraint setting a lower bound on the accepted *price-taking quantity* (see Figure 19 where the dotted line indicates the minimum of *price-taking supply quantity* to be accepted).

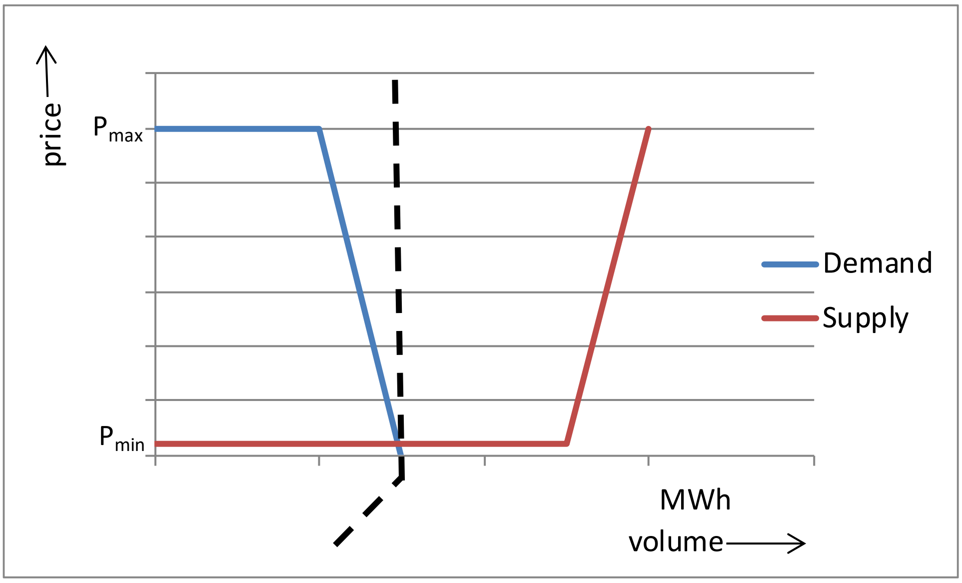


Figure 19 – Dotted line indicates the minimum of (*price-taking*) supply volume to be accepted

This constraint is referred to as the LOCAL_MATCHING constraint, and it is active in the master problem, i.e. prior to the price- and volume- coupling problems, but as an additional constraint to the economic surplus maximization problem.

7.9.2. Curtailment sharing

In this section we introduce the notion of the curtailment ratio, which is defined as the fraction of the price taking curve quantity that is rejected in each bidding zone:

$$curtailment\ ratio = 1 - \frac{accepted\ price\ taking\ volume}{submitted\ price\ taking\ volume}$$

The aim of curtailment sharing is to equalize as much as possible the curtailment ratios between those bidding zones that are simultaneously in a curtailment situation, and that are configured to share curtailment.

This curtailment sharing is implemented in part in the master problem and in part in the curtailment sharing volume problem step.

Curtailment Sharing – Master Problem[4]

The objective function of the master problem is to maximize economic surplus. For an ATC line this results in a situation where areas that are not in curtailment will export to areas that are in curtailment.

However, under FB this is not necessarily the case: if an exchange from area A to area B results in a higher usage of the capacity compared to an exchange A to C it is possible that is more beneficial to exchange from A to C, whereas market B is in curtailment. This is referred to as "flow factor competition".

In order to prevent such cases on demand side (effectively treating curtailment outside of the economic surplus maximizing framework) we penalize the non-acceptance of price taking demand orders (or PTDOs) by adding to the primal objective:

$$-M \cdot \sum_{h} MAX\_CURTAILMENT\_RATIO$$

Where:

MAX_CURTAILMENT_RATIO: the largest non-acceptance ratio of the price taking order across all areas

M: a large value, used as penalty

This expression is added to the economic surplus. If the value of M is sufficiently large, it will help minimize the rejected price-taking quantity in all markets, before looking for a solution with a good economic surplus. The infinite norm penalty function will tend to harmonize the curtailment ratios across the curtailed markets if any.

Curtailment sharing volume problem

For the case where areas were not affected by "flow factor competition", i.e. under ATC market coupling, curtailment sharing is targeted in the volume problem. Provided ATC capacity remains, the economic surplus

[4] This functionality was first available in EUPHEMIA 9.3 using a slightly different penalty function. The one presented in this document was adopted first in EUPHEMIA 9.4 (in production since 21 April 2016).

function is indifferent between accepting price taking orders of one bidding zone or another.

This step aims to equalize curtailment ratios as much as possible among *bidding zones* willing to share curtailment. Bidding zones that are not willing to share curtailment will have their curtailment fixed in the economic surplus maximizing solution where the LOCAL_MATCHING constraint prevented these areas to be forced to share curtailments. At the same time the LOCAL_MATCHING constraint of adjacent areas prevented non-sharing areas to receive support from sharing areas. The supply or demand orders within a *bidding zone* being in curtailment at maximum (minimum) price are shared with other *bidding zones* in curtailment at maximum (minimum) price. For those markets that share curtailment, if they are curtailed to a different degree, the markets with the least severe curtailment (by comparison) would help the others reducing their curtailment, so that all the *bidding zones* in curtailment will end up with more equal curtailment ratios while respecting all network constraints.

The curtailment sharing is implemented by solving a dedicated volume problem, where all network constraints are enforced, but only the acceptance of the price taking volume is considered in the objective function. The curtailment ratios weighted by the volumes of price taking orders is minimized:

$$\min \quad \sum_{h} \sum_{\substack{m: \\ m \in C_{h,Supply}}} \sum_{\substack{o: \\ market(o)=m \\ p_o = p_{\min,m}}} |q_o| \left(1 - ACCEPT_o\right)^2$$
$$+ \sum_{h} \sum_{\substack{m: \\ m \in C_{h,Demand}}} \sum_{\substack{o: \\ market(o)=m \\ p_o = p_{\max,m}}} |q_o| \left(1 - ACCEPT_o\right)^2$$

One can prove that for optimal solutions for this problem in the absence of any active network constraints this will result into equal curtailment ratios.

7.9.3. Maximizing Accepted Volumes

In this step, the algorithm maximizes the accepted volume.

All period orders, complex period sub-orders, merit orders and PUN orders are taken into account for maximizing the accepted volumes. The acceptance of most orders is already fixed at this point. Either because it is completely below or above the *market clearing price*, or it is a *price-taking order* fixed at the first or second volume indeterminacy sub-problem (curtailment minimization or curtailment sharing). Block orders are not considered in this optimization because a feasible solution has been found prior to this step in the master problem.

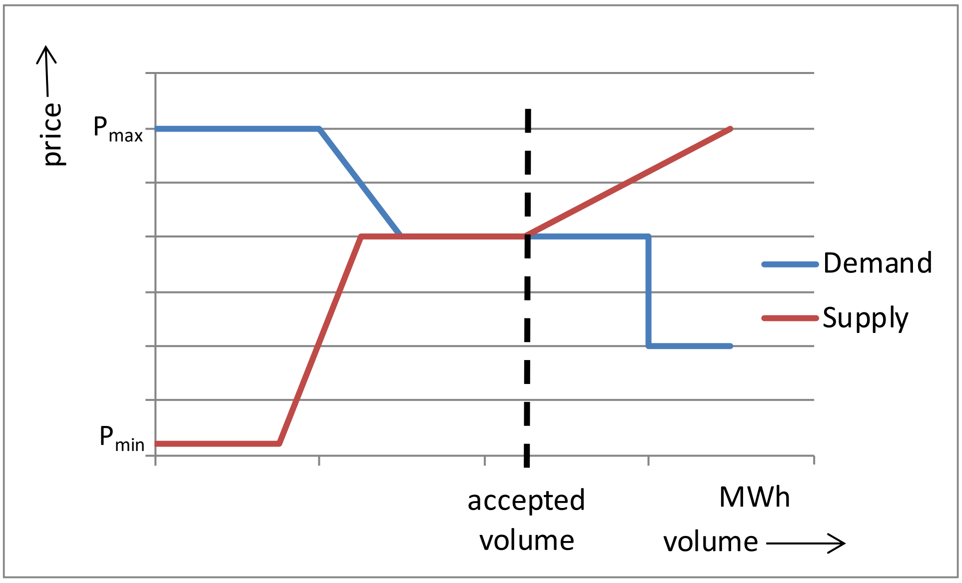


Figure 20 – The accepted volume is maximized

7.9.4. Merit order enforcement

This step enforces merit order numbers of the period orders if applicable. The acceptance of period orders with merit order numbers *at-the-money* is relaxed and re-distributed according to their acceptance priority. This problem is solved only if the solution found satisfies the PUN requirements (after the PUN search) or if there are no PUN orders but there exist some merit orders.

7.9.5. Flow indeterminacy

The last sub-problems re-attribute flows at the bidding zone, scheduling area and NEMO trading hub levels, to have fully determined rules. This section outlines the high-level principles that are applied. More details on the implementation can be found in the annexes.

Bidding Zone flow indeterminacy

At bidding zone level, scheduled exchanges between pairs of bidding zones are computed. Scheduled exchanges on the lines are based on the linear and quadratic cost coefficients of associate to these lines. Apart from the scheduled exchanges, all other variables are fixed to their predetermined value. This step can only affect the results in situations where there is full price convergence within a meshed network, allowing multiple flow assignments to result in identical *net positions*. By using specific values for the cost coefficients, certain routes will be chosen and unique flows will be determined.

Scheduling Area flow indeterminacy

Where the scheduling area equals to bidding zone then the same rules like for BZ scheduled exchanges shall apply. If there is more than one scheduling area in bidding zone, then scheduled exchanges between pairs of scheduling areas are computed, once bidding zone flows have been determined. In case of cross zonal scheduling area lines thermal capacity constraints are considered to distribute the bidding zone flows among the corresponding SA lines proportionally to their thermal capacities. In case of intra zonal scheduling area lines the SA scheduled exchanges are determined based on the linear and quadratic cost coefficients associated to each intra zonal scheduling area line. Similarly like in case of scheduling calculation at BZ level, by using specific values for the cost coefficients, certain routes will be chosen and unique intra zonal scheduling area flows will be determined.

NEMO Trading Hub flow indeterminacy

Once both inter zonal and intra zonal Scheduling Area flows have been defined, EUPHEMIA will compute the flow corresponding to each existing NTH line. Such flows are computed via the Inter-NEMO Flow Calculation (INFC) module, whose approach aims at minimizing the *net financial exposure* between each pair of Central Counterparties (CCPs) which manage the financial exchanges between NEMOs. The Annex C. on flow calculation models details the mathematical aspects of this minimization.

If any indeterminacies remain, these are resolved using linear and quadratic cost coefficients associated to each of the NEMO trading hub lines.

Degraded mode

Numerical difficulties might happen (at least in theory) during the SA flow determination or during INFC, as these are themselves based on optimization problems. For such cases, a fallback flow determination approach has been designed in order not to discard a valid market coupling solution. It is based on simple heuristics which will provide sub-optimal solutions, but solutions which are still valid with regards to the business constraints.

See 0 for more details on the details relative to the degraded mode implementation.

7.10. Multi-threading approach

In order to improve the quality of the solution (i.e. the limit the number of PRBs), to increase economic surplus and to anticipate future market evolutions, by EUPHEMIA 10 introduces a multi-threading approach.

In particular, EUPHEMIA 10 has been designed according to the following architecture:

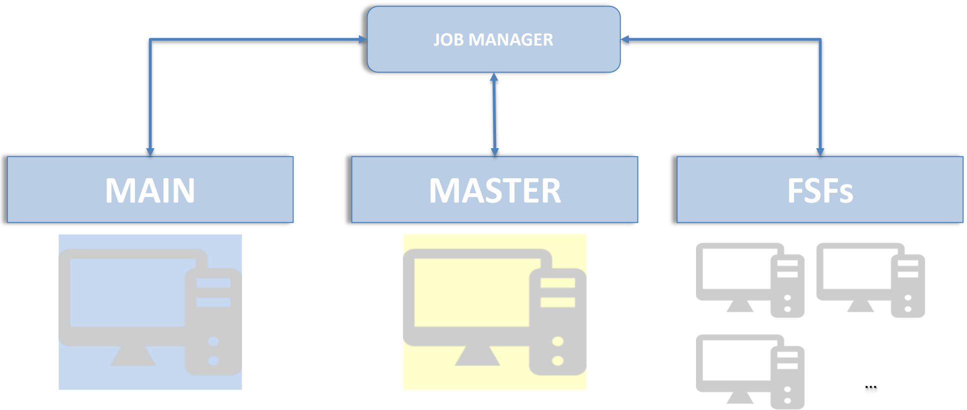


Figure 20 – EUPHEMIA 10 architecture

- **Job Manager**: It's not a thread, but a data structure. It connects different actors of the process.
  In particular:
  - It starts Master's activity.
  - It sends Master solutions to FSFs.
  - It receives the valid solution by FSFs.
  - It triggers the FSFs' local search.
- **Main Core**: The main thread checks the input data, initializes the data model and creates the computation threads. It waits for the computation threads to finish their tasks and writes the the last information in the database before terminating the run.
- **Master Core**: On the basis of the input provided by MAIN, MASTER solves the primal problem (first volumes calculation, not prices and not the final volume problem). In order to do that, it performs branch and bound to find not partially accepted node. Every time it finds an entire node (selection of fully accepted/rejected MICs and blocks), it sends information to FSFs.
- Since the input provided by MASTER, **a Feasible solution finder (FSF)** solves the dual problem (prices calculation, PUN search, no Paradoxically accepted MICs, no Paradoxically accepted blocks, flows intuitiveness…). Different Alternative Configuration can be used per different FSF, in order to increase possibilities to find a feasible solution. As soon as a valid solution is found, FSF performs local search to improve it (PRBs reinsertion).

The different tasks performed by the different actors are coordinated according to the figure below:

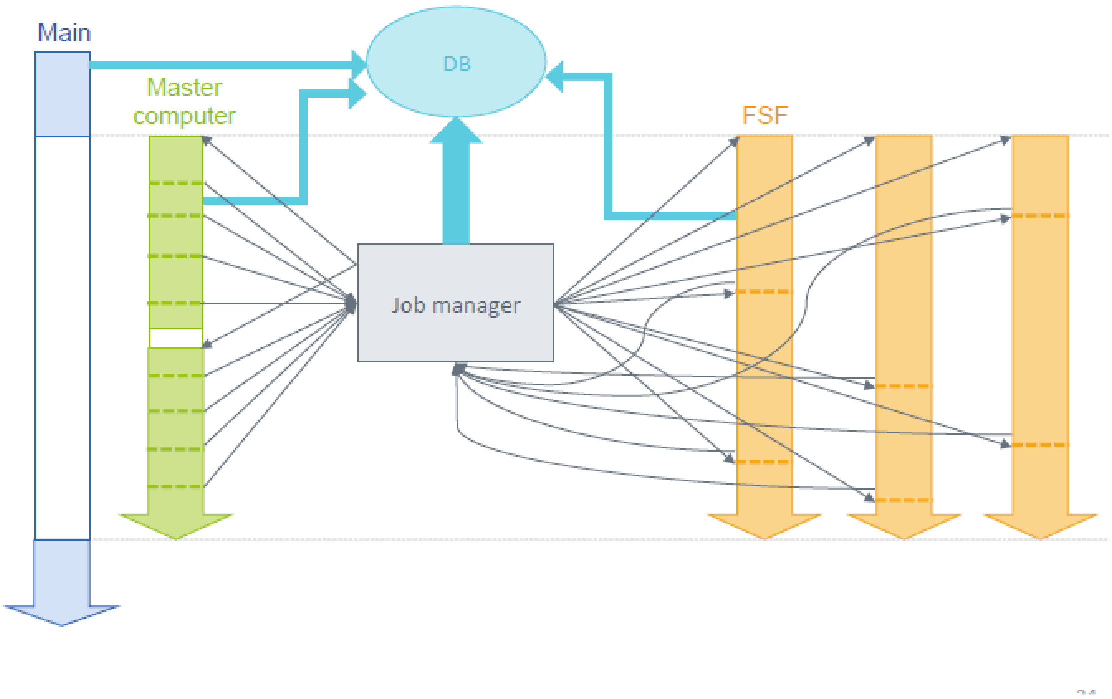


Figure 21 – Interaction actors

In EUPHEMIA threads do not communicate directly with each other but use the job manager for passing information, e.g. FSFs receive FSF jobs (e.g. all-kill jobs) from job manager and cut off values and return solutions).
For this reason, it is not granted that two different runs, on the same machine, will return exactly the same results.
A possible solution to ensure repeatability is to use the deterministic time approach:

- Jobs are assigned to FSFs deterministically.
- Jobs assignment becomes independent on the speed of the FSFs.

7.10.1. Repeatability

The repeatability of an algorithm is defined as the capability of the algorithm to reproduce the same results upon request. On the same machine, two subsequent runs with the same input data should find the same solutions, meaning that the intermediate/final solutions found at iteration 'X' are the same. In other words, when the stopping criterion is the number of investigated solutions, a reproducible algorithm can guarantee to obtain the same final result when run on the same machine. However, when the stopping criterion is a time limit, a faster computer will allow the algorithm to investigate more solutions than a slower one. In this case, the repeatability consists in investigating on the faster computer at least the same set of solutions as the ones investigated on the slower computer.

Mind that with the introduction of the PRB reinsertion (cf. section 6.7), another time limit is introduced: the PRB reinsertion process times out too, ahead of the final time limit. This should therefore be understood as a time limit in its own right and repeatability only applies up until this point.

Deterministic timing is a measure for improving repeatability of EUPHEMIA in different runs on the same input data. The measure is covering following causes of non-repeatability:

- Multithreading;
- time limit;
  - used by local search;
  - used by CPLEX for solving sub-problems.

7.10.2. Deterministic Clock

Algorithm computation can take a different amount of time in two different runs, thereby e.g. the same sub-problems limited by the same time limit may end up with a different solution in both runs. The remedy for this may be a deterministic clock usage to measure time in ticks, which are normally consistent measures for a given platform (combination of hardware and software). Multithreading can use deterministic clock to measure deterministic time (deterministic time describes a measure for the amount of computation performed by the thread) and to synchronize the deterministic clocks of the threads whenever information is exchanged between them.
A deterministic time reported by CPLEX is used as a deterministic time for EUPHEMIA.

7.10.3. Remedy for Non-repeatability Caused by Multithreading

In EUPHEMIA deterministic clocks of threads are synchronized by mechanism called synchronization barrier - whenever a thread visits the synchronization barrier it must wait until all other threads arrive. Once the barrier is reached by all threads, FSFs can start their computation again in a non-synchronized way until they reach the synchronization barrier again. Each thread measures deterministic time passed since last synchronization barrier (or since the beginning of the computation if no synchronization happened before) to determine when to stop at the synchronization barrier (barriers distance is specified amount of deterministic time).

7.10.4. Remedy for Non-repeatability Caused by Local Search

Local search is a measure to try to improve the solution by reinserting rejected blocks or swap active and non-active blocks. To ensure repeatability, the sequence in which improving solutions from local search are inserted, shall be identical between runs. To avoid this sequence is altered due to differences in machine load, this too shall be managed by the synchronisation events governed by “deterministic time”.
In EUPHEMIA when a thread running local search visits the synchronization barrier and better solution found by the local search is available, then the

solution is written. In this way, it is guaranteed that the local search finds the same solutions during different runs.

7.10.5. Remedy for Non-repeatability Caused by Sub-problems

Sub-problems in CPLEX use time limits. To achieve repeatability, it is important to use deterministic time limits, to avoid sub-problem feasibility being dependent on clock time leading to a situation where some problem might appear infeasible for one run but might be solved in another run because the computation is faster.
EUPHEMIA is using a feature of CPLEX, which offers the possibility for such deterministic time limits to be used by sub-problems, so it is guaranteed that sub-problems interrupt the computation at the same stage during different runs.
The bidding zone flow calculation as governed by the TSO SEC methodology introduces another source of non-reproducible behaviour: in case the problem takes "too" long this change allows Euphemia to automatically trigger a simplified (linearised) version of the problem. The trigger will be time based, hence not repeatable. As long as the fallback is not triggered, deterministic time should still allow the calculation to be repeatable.

7.11. Stopping criteria

As an optimization algorithm, EUPHEMIA searches the solution space for the best feasible solution until some stopping criterion is met. The solution space is defined as the set of solutions that satisfy all the constraints of the problem.

EUPHEMIA is tuned to provide a first feasible solution as fast as possible. However, after finding the first solution, EUPHEMIA continues searching, the solution space for a better solution until a stopping criterion for example the maximum time limit of 12 minutes, is reached or until a feasible selection of blocks and MIC orders no longer exists.
The calculation will stop either when the full branch and bound tree is explored or when one of these criteria is reached:

- TIME LIMIT
  This parameter sets a limit to the total running time of EUPHEMIA. However, since the time taken by operations after calculation (e.g. writing of the solution in the database) can be variable, this is an approximate value.

  In case the time limit is reached, but no valid solution is found, the calculation continues and stops only when a first solution is found. A second-time limit applies for finding this first solution: if it times out the session fails and EUPHEMIA does not return any solution.

- ITERATION LIMIT

EUPHEMIA can stop after it has processed a given number of nodes. In SDAC the iteration limit is not used as a stopping criterion.

o SOLUTION LIMIT
EUPHEMIA can stop after it has found a given number of solutions (regardless of their quality). In SDAC the solution limit is not used as a stopping criterion.

8. Additional Requirements

8.1. Precision and Rounding

EUPHEMIA provides results (unrounded) which satisfy all constraints with a target tolerance. These prices and volumes (flows and *net positions*) are rounded by applying the commercial rounding (round-half-up) convention before being published.

8.2. Solution validation

Once a feasible solution has been found, EUPHEMIA explicitly verifies all properties and constraints:

- Constraints. Define the gap as:
  - the difference between the left-hand side and right-hand side for less-than constraints;
  - the difference between the right-hand side and the left-hand side for greater-than constraints;
  - the absolute value of the difference between the right-hand side and the left-hand side for equalities;
- Bounds on variables. The gap for an upper (respectively lower bound) follows the same approach as a less-than (respectively greater-than) constraint.
- Integrality
  - These checks confirm the integrality of the binary variables. The gap is defined as the absolute difference between the value of the variable and its rounded integer value.
- Complementary slackness
  - Those conditions correspond to the price-network properties (e.g. no price difference without congestion) and the order acceptance criteria (e.g. no paradoxically accepted/ rejected curve orders).
  - They connect the fact that a shadow price (e.g. a congestion rent) is positive when the corresponding constraint on the volumes is tight (e.g. the ATC constraint

on the flow). This relationship is justified by the optimality of the economic surplus of the solution for a fixed selection of block orders, complex orders and PUN volume. Essentially, any solution that would not satisfy its complementarity conditions could be improved by setting the shadow price to zero or tightening the constraint.

In absence of tariffs, losses and ramping, the market prices on two bidding zones can be different if and only if all their capacities for (freely) exchanging energy are saturated (i.e. ATC and PTDF capacities in a Hybrid model).

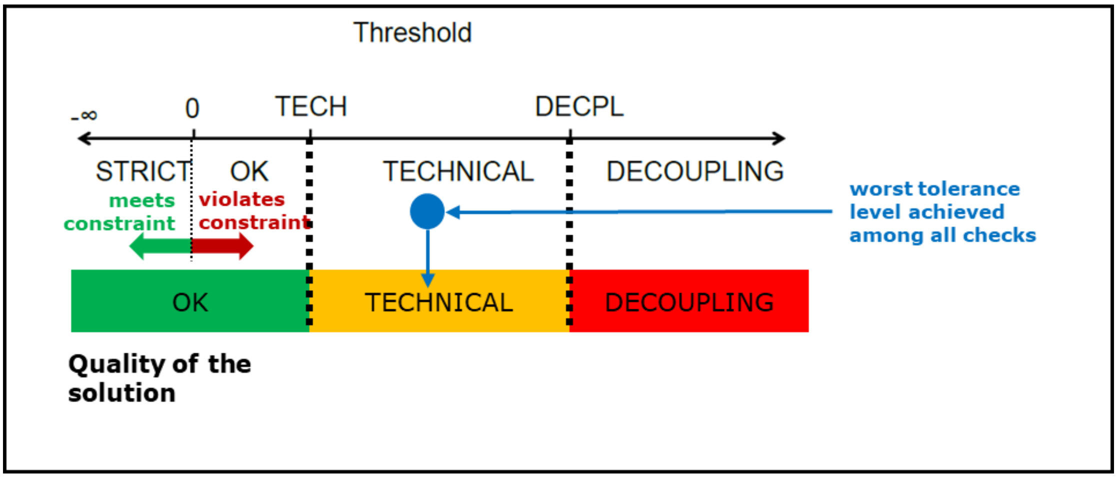
Indeed, if it was possible to send ε MW more from a bidding zone A to B where the price is Δ €/MWh higher, doing so would substitute lower-priced demand from A (and/or higher-priced supply from B) by higher-priced demand from B (and/or lower-priced supply from A), for a welfare benefit of ε Δ. This can be done until the prices converge, or there is no more capacity for exchange.

- Here since either the shadow price must be zero or the constraint must be tight, the gap is defined as the minimum between the value of the shadow price and the slack of the constraint.

The various checks are performed against tolerance thresholds. For each constraint type, two levels of tolerance are defined: a TECH level and a DECPL level with 0 < TECH < DECPL. These tolerances are used to qualify each check:

- STRICT: No violation of constraints (violation strictly 0)
- OK: Acceptable violation of constraints (violation within technical limit)
- TECHNICAL: Mild tolerance violation (violation within decoupling limit)
- DECOUPLING: Severe tolerance violation (violation exceeding decoupling limit)

We can illustrate the link between the tolerance threshold that is respected and the achieved solution quality using:



The overall quality of a given solution is determined by the worst tolerance level achieved among all checks. Regarding solution quality, both STRICT and OK are reported as OK solutions.

Properties of the solution

During the execution of EUPHEMIA, several feasible solutions can be found. However, only the best solution found before the stopping criterion of the algorithm is met is reported as the final solution, where best means:

- Solution with the highest quality (- OK > TECHNICAL > DECOUPLING)
- Solution with the highest primal objective function value
    - This is the objective function that includes the curtailment penalty terms. In the absence of curtailment this is identical to the economic surplus. For curtailment cases there will be a difference, and this objective shall be considered first, or the solution selection could counteract the objectives of the curtailment mitigation.
- with the highest economic surplus value (complying to all network and market requirements)

It should be noted that for difficult instances some heuristics[5] are used by EUPHEMIA in its execution. Thus, it cannot be expected that the "optimal" solution is found in all cases.

[5] In mathematical optimization, a **heuristic** is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. This is achieved by trading optimality, completeness, accuracy, and/or precision for speed (Ref-: http://en.wikipedia.org/wiki/Heuristic_(computer_science)).

8.3. Transparency

EUPHEMIA produces feasible solutions and chooses the best one according to the agreed criterion (economic surplus-maximization). Therefore, the chosen results are well explainable to the market participants: published solution is the one for which the market value is the largest while respecting all the market rules.

Annex A. Glossary

Item	Acronym	Description
Already Allocated Capacity	AAC	means the total amount of allocated transmission rights, whether they are capacity or exchange programmes depending on the allocation method;
Adverse Flow		In market coupling, it is expected that the flow between two bidding zones goes from the market with a lower price towards the market with a higher prices. However, it may happen that, due to some constraints such as the ramping constraint imposed on some interconnectors, the cross-border flow end up being, at some particular periods, in the direction from a higher price bidding zone towards a lower price bidding zone. These flows are commonly known as "Adverse flows" and force the Congestion Rent to be negative.
At-the-money		A supply (demand) order is considered at-the-money if its price is equal to the market clearing price. If the time resolution of the order is different than the MTU of the bidding zone, the market price to consider is the arithmetic mean of the underlying market prices. For blocks this notion is generalized by considering the volume weighted average price.
Bidding zone	BZ	A bidding zone is a geographical area to which network constraints are applied. Consequently all submitted orders in the same bidding zone will necessarily be subjected to the same unique price.
Congestion Rent		In an ATC model, the Congestion Rent measures for each interconnector traversed by a flow the difference between the total amount of money to be paid to the supplier of this flow at one end of the interconnector (market clearing price of the supplying bidding zone × the volume of the energy flow through the interconnector) and the total amount of money to be received from the consumer of this flow at the other end of the interconnector (market clearing price of the consuming bidding zone × the volume of the energy flow through the interconnector). It is equal to the product of the cross-border price spread and the implicit flow obtained by EUPHEMIA. The presence of losses on the interconnector will not impact the congestion rent. However, if the interconnector implements tariffs, the congestion rent will be reduced by the product of the tariff rates and the implicit flow obtained by EUPHEMIA.
Consumer Surplus		The Consumer Surplus measures for the buyers whose orders are executed the difference between the maximum amount of money they are offering (limit price of their order × the executed volume of their order) and the amount of money they will effectively pay (market clearing price × the executed volume of their order).
Critical Network Element	CNE	

Day-Ahead	DA	The DA market is a market for electricity trading with delivery of physical power period-by-period the next day.
Deep in the money		A supply (demand) order is considered In-the-money if its price is smaller (greater) than the market clearing price plus a specified parameter (Max Delta P).
False paradoxically deactivated complex MIC orders		A false paradoxically deactivated MIC order (false PR MIC) is a deactivated MIC whose economic condition seems to be fulfilled with the MCPs obtained in the final solution (so it seems that it should be activated) but, after acceptance its economic condition is not fulfilled anymore.
Flow Based	FB	The Flow Based (FB) model is an alternative to ATC network constraints
Intra Day Auction	IDA	intraday auction' (IDA) means the implicit intraday auction trading session for simultaneously matching orders from different bidding zones and allocating the available intraday cross-zonal capacity at the bidding zone borders by applying a market coupling mechanism;
Interconnector		Transmission line which crosses or spans a border between countries and which connects the national transmission systems of the countries;
In-the-money		A supply (demand) order is considered in-the-money if its price is smaller (greater) than the market clearing price. If the time resolution of the order is different than the MTU of the bidding zone, the market price to consider is the arithmetic mean of the underlying market prices. For blocks this notion is generalized by considering the volume weighted average price.
Line		An abstract representation that connects two bidding zones;
Long Term Allocation	LTA	The LTA domain includes the long-term capacities allocated explicitly which are offered for some borders.
Market Clearing Price	MCP	A common reference price for the whole Market area, when not considering transmission constraints.
Market Time Unit	MTU	The period for which the market price is established.
Minimum Income Condition	MIC	The Minimum Income Condition (**MIC**) (respectively Maximum Payment condition (**MP**)) in complex orders and/or scalable complex orders adds an economic condition to sell complex order (respectively, buy complex order), which represents the minimum income (respectively, the maximum payment) expected, by order's owner defined by a fix term in euros or/and, in the case of complex orders only, a variable term in euros per accepted MW produced (consumed, respectively) for the set of curve sub-orders.
Maximum Payment condition	MP	
NEMO Trading Hub	NTH	NEMO trading hub – combination of NEMO, active in a scheduling area, within a bidding zone
Net position (net export position)		The difference between accepted local supply and demand for a bidding zone.
Out of the money		A supply (demand) order is considered out-of-the-money if its price is greater (smaller) than the market clearing price. If the time resolution of the order is different than the MTU of the bidding zone, the market

		price to consider is the arithmetic mean of the underlying market prices. For blocks this notion is generalized by considering the volume weighted average price.
Paradoxical acceptance of block orders	PAB	A block which is accepted while being out-of-the-money.
Paradoxical rejection of block orders	PRB	A block which is rejected while being in-the-money or at-the-money
Power Transfer Distribution Factor	PTDF	When FB model is used the electricity market receives a linearized “security domain” described by PTDFs on CNEs.
Price-taking order	PTO	buy at max price of sell at min price period orders submitted at the MTU of the bidding zone.
Producer Surplus		The Producer Surplus measures for the sellers whose orders are executed the difference between the minimum amount of money they are requesting (limit price of their order × executed volume of their order) and the amount of money they will effectively receive (market clearing price × executed volume of their order).
Remaining Available Margin	RAM	Maximum flow minus the flow in the base case including long term capacities and minus the flow reliability margin; RAM specifies the free margin for every cross-border.
PUN price		PUN is the average (weighted by purchased quantity of PUN orders) of GME Zonal Market Prices (Italian "physical" zones). PUN is the price to consider accepting/rejecting purchase period orders made by PUN orders (“consumption purchase period orders”).
Scheduling Area	SA	Delivery area within a bidding zone, i.e. typically an area under the control of a TSO
Transmission System Operator	TSO	Transmission System Operator
Virtual bidding zone		Virtual bidding zone is a zone with empty nemo hubs without orderbooks. Each virtual bidding zone has a dedicated parent bidding zone. If so desired, the TSO of the virtual BZ may decide that rounding residuals from virtual bidding zones shall be sent to their parent bidding zone.

Annex B. Heuristics

EUPHEMIA relies on some heuristics to manage with the European coupling problem. This section outlines some of the heuristics on which EUPHEMIA relies.

Blocks and MICs -> Branch A/B

When the price problem has no solution, we know that the block and complex order selection has made the problem infeasible, so the integer

solution to the Master Problem must be rejected. To do so, we add a local cut in the primal problem that makes the current solution infeasible.
For each type of additional requirement to be satisfied by the prices, a specific type of cut is created aiming at enforcing compliance with the corresponding requirement:

- Block cut: When the no PAB requirement is not satisfied by the selected block orders.
- MIC cut: When the minimum income or maximum payment conditions are not satisfied by the selected complex orders.
- Intuitive cut: When the flows are not intuitive.

Consider the case of a problem that deals just with blocks, and the no PAB requirement is not satisfied, since no prices exist without some accepted blocks being out-of-the-money. In this case we cannot just reject the PABs. The goal is to kill other blocks, so that the prices will change and the block will no longer be PAB.

Thus, we create a new node where a cut is added, which invalidates the current block selection. In this new node, the cutting method has to reject blocks that are in-the-money. Bearing in mind that the algorithm objective is economic surplus optimization, EUPHEMIA might reject blocks with low volumes that are deep in-the-money, because the impact of these blocks on economic surplus might be negligible. To prevent this unwanted behaviour, only the blocks that are in-the-money by less than a threshold parameter named delta parameter will be introduced in the cut. This parameter prevents the rejection of block orders that are in-the-money by more than delta parameter monetary units. By introducing this parameter, we use a heuristic, as all in-the-money blocks should be considered if we want to avoid cutting off some of the search space prematurely.

When only complex orders with minimum income condition exist, a similar approach is followed.

In the cut strategy heuristic, only the orders below a threshold named delta parameter are considered for the application of the cut. This heuristic parameter prevents the rejection of block or complex orders that are in-the-money by more than delta monetary units. A careful selection of delta parameter should be done for tuning the heuristic. If it is too low, we select less block or complex orders and we are being more aggressive on the blocks or complex orders that are closer to their acceptance conditions, potentially cutting better solutions.
A large value of delta parameter may bring us closer to the optimal solution (because we are less aggressive), albeit at the cost of algorithm time performance. But it will allow deep-in-the-money PRBs (Paradoxically rejected block orders).

When the algorithm deals with both blocks and MIC or MPC orders, we associate both cutting strategies in a combined soft cut strategy. This soft cut strategy is defined by two branches:

- Under Branch A, one of the least promising accepted block or complex orders must be rejected. Performing branch A repeatedly at each block or MIC cut always leads to a feasible solution, because rejecting blocks and deactivating MICs and MPCs eliminates orders violating their condition. Therefore, Branch A is more likely to rapidly give a solution and should be explored first.

- Branch B is complementary to branch A and its creation is optional. It is not as likely to give a solution because it forces the acceptance of the least promising block orders and/or complex orders. There are heuristics in complex orders in which in certain situations branch B is not explored, however, the solution could have a larger gap (the difference between the best bound provided by the solver at the time the solution was found and the utility of the solution) as a result of not exploring nodes in this branch.

Annex C. Mathematical Approach

Purpose of EUPHEMIA algorithm is to grant the maximization of economic surplus, under a set of given constraints:

- network constraints
- clearing constraints
- period order acceptance rules
- price network properties
- kill – or – fill conditions
- no PAB constraints
- MIC constraints
- PUN consecutiveness constraints
- PUN imbalance constraints

In order to pursue this issue, EUPHEMIA relies on the concept of duality[6] to calculate prices and volumes on which economic surplus calculation is based on.

[6] Duality is a relationship between two problems, called respectively the primal and dual. Each constraint in the primal problem corresponds to a variable in the dual problem (called its dual variable), and each variable in the primal problem has a corresponding constraint in the dual problem. The coefficients of the objective in the dual problem correspond to the right-hand side of the constraints in the primal problem. When the primal problem is a maximization problem, the dual is a minimization problem and vice-versa. Linear optimization problem is the dual of its dual. In the case of a convex problem, duality theory states that if both primal and dual problems are feasible, the optimal solutions of the primal and dual problems share the same objective value and exhibit a special relationship, called complementary slackness conditions. Specifically, whenever a constraint is not binding in the optimal primal (resp. dual) solution, then the corresponding dual (resp. primal) variable has a value of zero in the optimal dual (resp. primal) solution. Conversely, when a variable has a non-zero value in the primal (resp. dual), the corresponding constraint must be binding in the dual (resp. primal).

In the case of EUPHEMIA, the primal and dual problem can be synthesized as follows:

Problem	Unit	Variables	Constraints
Primal	MWh	Acceptance of Order Flow between bidding zones	Precedence between orders Network load limitations
Dual	€/MWh	Market Clearing Prices Congestion Rent	Constraints on price differences

Strictly speaking, there are some reasons why the primal and dual problems in EUPHEMIA do not fit exactly in the above duality context.

1. The objective of the primal problem (the economic surplus) is quadratic in terms of the acceptance variables. This is due to the interpolated orders: their marginal contribution to the economic surplus varies with the proportion matched. Fortunately, the Lagrangian duality principle still applies in the context of problems with quadratic objectives.
2. The primal problem contains integer variables. This is due to the presence of binary variables to represent the activation of blocks and complex orders. The linear duality theory unfortunately does not extend immediately to problems with integral variables. However, as soon as all integer variables have been fixed to certain values (that is, for a given selection of blocks and complex orders), then we are back into the regular duality theory context.
3. The dual problem in EUPHEMIA contains additional constraints which do not emerge naturally from the primal problem[7].
4. The coupling problem involves so called primal-dual constraints, i.e. constraints involving both primal and dual variables in their expression[8].
5. Not all dual variables are created. In particular, each order acceptance variable is bound to 1. This constraint should normally have a dual surplus variable, which would then play a role on the admissible prices. Almost all of those constraints would be redundant, so in the dual model of EUPHEMIA the price bounds are computed explicitly, and the surplus variables are not created.
6. The objective of the dual problem used by EUPHEMIA does not correspond to the primal one. Indeed, the objective value is already

[7] For example: the condition of accepted blocks to be not paradoxically accepted is not naturally met by an optimal primal-dual solution. Intuitively, this is related to the integer nature of the primal problem: by imposing the selection of blocks, we are exposed to the fact that some are losing money individually for the benefit of the economic surplus.

[8] For example, the Minimum Income Condition for complex orders involves both the volumes matched (i.e. primal variables) and the market clearing prices (i.e. dual variables). Those constraints can only be formulated in the dual problem by substituting the corresponding primal variables by their optimal value in the primal problem, and reciprocally in dual one.

known from the primal problem and the goal of the dual problem will be to tackle other requirements, e.g. price indeterminacy rules.

Annex C.1.Economic surplus Maximization Problem

The purpose of the Master Problem is to find a good selection of blocks and complex orders (i.e. all binary variables) satisfying all of their respective requirements. The objective function of this problem is to maximize the global economic surplus:

$$- \sum_{\substack{z,t,s,o: \\ Step\ Orders}} ACCEPT_{z,t,s,o} q_{z,t,s,o}\ p^{O}_{z,t,s,o} res(o)\ \ (1)$$

$$- \sum_{\substack{z,t,s,o: \\ Interpolated\ Orders}} ACCEPT_{z,t,s,o} q_{z,t,s,o} \Bigg(p^{O}_{z,t,s,o} + ACCEPT_{z,t,s,o}\ \frac{p^{1}_{z,t,s,o}\ - p^{O}_{z,t,s,o}}{2} \Bigg) res(o)\ (2)$$

$$- \sum_{bo,t} ACCEPT_{bo} q_{bo,t}\, p_{bo} res(o)\ \ (3)$$

$$- \sum_{z,co,t,o} ACCEPT_{z,co,t,o}\ q_{z,co,t,o} p_{z,co,t,o} res(co)\ (4)$$

$$- \sum_{z,sco,t,o} ACCEPT_{z,co,t,o}\ q_{z,co,t,o} p_{z,co,t,o} res(sco) - sign(type(sco) FixedTerm_{sco} B\_ACCEPT_{sco}\ (5)$$

$$- \sum_{mo} ACCEPT_{mo}\ \ q_{mo} p_{mo} res(mo)\ \ (6)$$

$$- \sum_{l,u,t} Tariff_{l,t}\ FLOW_{l,u,t}\ \ (7)$$

$$-M \sum_{\substack{z,t,o: \\ Price\ Taking\ Hourly \\ orders}} MAX\_CURTAILMENT\_RATIO\ \ \ (7)$$

where (bearing in mind that q_o is positive for a supply order and negative for demand orders):

1. is the contribution of period step orders
2. is the contribution of period interpolated orders
3. is the contribution of block orders
4. is the contribution of complex orders
5. is the contribution of scalable complex orders
6. is the contribution of merit orders

7. is the impact of Tariffs
8. This expression is added to the economic surplus. If the value of M is sufficiently large, it will help minimize the rejected price-taking quantity in all markets, before looking for a solution with a good economic surplus. The quadratic function will tend to harmonize the curtailment ratios across the curtailed markets if any

Mind that for the price taking orders, only the orders submitted at the MTU of the bidding zone are considered. E.g. for a 15' bidding zone with a maximum price of 4000€/MWh, a 30' buy period order at 4000 is **not** considered as price taking.

Subject to:

- Market constraints
  - Balance/clearing constraints
  - Block order acceptance constraint
  - Complex suborders acceptance constraints
  - Load Gradient constraint
  - Merit order acceptance constraints
- Network constraints
  - ATC constraints
  - PDTF constraints
  - Various ramping constraints

Annex C.2.Price Determination Sub-problem

For each feasible solution of the primal problem, EUPHEMIA solves the following price problem:

$$\min_{prices} distance\ to\ mid\ point$$

i.e.:

$$min \sum_{m,h} \left(MCP_{m,h} - \frac{UpperBound_{m,h} + LowerBound_{m,h}}{2} \right)^2$$

Subject to:

- complementarity slackness conditions
- price bounds
- no PAB constraints
- Minimum Income Condition
- PUN imbalance

Price indeterminacies are resolved differently for "Satellite bidding zones". A zone is defined to be a satellite bidding zone for period t if it satisfied the following conditions:

- t has the same time resolution as the one of the bidding zone
- One single curve with power> 0 at period t (supply or demand)
- On the parent periods of t no power is offered in any direction by any type of orders
- No block orders defined at period t
- No PUN orders at period t
- No merit orders at period t
- No LG orders (at any period)
- No complex suborders at period t
- No daily market ramping
- No period market ramping at period t
- A single ATC connection, which:
    - is not included in any line set
    - have the same time resolution as the bidding zone
    - does not have losses at period t
    - does not have tariffs at period t
    - does not have ATC ramping at period t
    - does not have AAC at period t
- A neighbour zone with the same time resolution, with no daily market ramping and no periodic market ramping at periods t and t + 1.

with only simple period orders of one type, all supply or all demand (including PTOs), that is connected with a single ATC line with the rest of the topology, no losses, no tariff, no ramping, doesn't participate to price determination sub-problem. When all the submitted volume is matched and equal to the ATC value the price in the satellite bidding zones will be set to the price of the adjacent bidding zone.

Annex C.3.Flow calculation models

This section outlines the different flow calculations models that EUPHEMIA solves, to uniquely establish scheduled exchanges at the bidding zone, scheduling area and NEMO trading hub levels respective. See section 7.9.5.

These models shall be compatible with the eventual TSO DA Scheduled Exchanges Calculation Methodology.

Bidding Zone (BZ) flow calculation

This step aims at uniquely define the flow results between bidding zones, in case indeterminacies remain. Several flow routes might be possible for given net positions. Flow calculation uses linear and quadratic cost coefficients associated to each of the BZ lines: EUPHEMIA minimizes the following function:

$$\min\left(\sum_{t}\sum_{l} res(l)\left(lc_l * \left(f_{l,t,up} + f_{l,t,down}\right) + qc_l * \left(f_{l,t,up}^2 + f_{l,t,down}^2\right)\right)\right)$$

Where *t* represents the periods, *l* the lines (both ATC and FB), res(l) time resolution of the period t expressed in hours (i.e., if the time-resolution of the line *l* is in minutes *r(l)* then the *res(l)=r(l)/60*), *up* and *down* the direction of the line, *lc* and *qc* the linear and quadratic cost coefficients of a line, and *f* the flow variables to be determined.

Scheduling Area flow calculation

The objective function of scheduling area flow calculation model is comparable to the one from the BZ flow calculation, but here the flows (or exchanges) between scheduling areas are considered when minimizing linear and quadratic flow function:

$$min\sum_{sl}\sum_{t\in T(sl)}\frac{1}{|T(sl)|}\left[lc_{sl,t,up}\left(f_{sl,t,up} + f_{sl,t,down)}\right) + qc_{sl}(f_{sl,t,up}^2 + f_{sl,t,down}^2)\right]$$

Where *t* represents the periods, *sl* the Scheduling Area lines, *up* and *down* the direction of the line, *lc* and *qc* the linear and quadratic cost coefficients of the line, and *f* the positive flow variables to be determined. *T(sl)* the set of periods *t* in the time resolution of scheduling area line *sl.* The division by |*T(sl)*| ensures fairness between lines of different time resolutions.

Calculation of Scheduled Exchanges between NEMO trading hubs

1. The Scheduled Exchange Calculator shall calculate the Scheduled Exchanges between NEMO trading hubs based on NEMO trading hubs' net positions.
2. The calculation of Scheduled Exchanges between NEMO trading hubs aims at minimizing the Net Financial Exposure (hereinafter referred to as "NFE") between the central counter parties associated to each NEMO (hereinafter referred to as "CCP"). The NFE between two pairs of CCPs is expressed with relation to the Scheduled Exchanges between the NEMO trading hubs of their corresponding NEMO as follows:

$$NFE_{A|B} = \sum_{t \in \mathrm{T}} \sum_{l \in L_{A,B}} P_B^h * flow_{n_1,n_2}^t - P_A^t * flow_{n_2,n_1}^t$$

with:
• A, B being two different CCPs
• $L_{A,B} = \{l = (n_1, n_2) \in L^d \mid ccp(n_1) = A \wedge ccp(n_2) = B\}$ being the set of all lines linking NEMO trading hubs of NEMO corresponding to CCP A and NEMO trading hubs of NEMO corresponding to CCP B. L^d is the set of all directed lines connecting two NEMO Trading Hubs.
• $ccp(n_1)$, $ccp(n_2)$ is a function giving the CCP corresponding to NEMO trading hub n_1 and n_2 respectively
• P_A^t, P_B^t is the clearing price for bidding zone of CCP A and B respectively for market time unit t
• $flow_{n_1,n_2}^t$ is the Scheduled Exchange from NEMO trading hub n_1 to NEMO trading hub n_2 for market period t
• t is the period and T is the set of all periods
The net financial exposure $NFE_{A|B}$ of a CCP A with regards to a CCP B expresses the financial risk that B will induce on A. As can be seen, it is netted over all BZs and periods. A net financial exposure can either be positive or negative. Also, it can be shown that $NFE_{A|B} = -NFE_{B|A}$ (therefore, as soon as it is non-null, they shall have opposite signs). The sum of all net financial exposures among all pairs of CCPs shall always be zero (financial balance).

3. The NFE is minimized using a linear approach
NFE is minimized and nemo flows are determined in a single optimization problem. This is done by minimizing the following objective function,

$$\sum_{\{c,c\prime\} \in CCP} \left| NET\_EXPOSURE_{c,c\prime} \right| + \alpha \sum_{nl} \sum_{t \in T(nl)} res(nl) \left[FLOW_{nl,p,up} + FLOW_{nl,p,down} \right] + \sum_{sa} \sum_{t \in T(sa)} \max_{nl \in NHL(sa), dir \in \{up,down\}} res(nl) FLOW_{nl,p,dir}$$

$(= Absolute\ exposures + \alpha\, Volume\ penalty + MinMax)$

Where *CCP* is the set of all CCPs, *nl* denotes nemo hub line, *sa* a scheduling area, *T(nl)* the set of periods corresponding to time resolution of nemo hub line, *T(sa)* a set of periods corresponding to time resolution of scheduling areas, α corresponds to the INFC_COEFF divided by the NET_EXPOSURE_SCALING_FACTOR, *NHL(sa)* the set of nemo hub lines inside a scheduling area *sa* (no cross-border line).

The first term minimises the sum of the absolute net exposures. It is however not enough to ensure a unique solution. Two other terms are thus added: volume penalty, which avoids loops and MinMax which distributes the flows as equally as possible.

Degraded mode

The first step computes the "inter-BA" SA and NTH flows. Given the SA line thermal capacities, the flows on the BA lines are split among the SA lines. Then the flow on each SA line is assigned to the corresponding NTH line with the smallest linear cost-coefficient. In case there exist more than one NTH line with the same lowest linear cost coefficient, the flows are split equally.

The second step computes the "intra-BA" SA and NTH flows. This step will be applied to all bidding zones separately. We use the term inner-BA net position to describe the value of the NTH net position increased by the incoming flows on inter-BA NTH lines and decreased by the outgoing flows on inter-BA NTH lines.

The heuristic computes the flows on intra-BA NTH lines by solving a minimum-cost maximum flow problem. To model the problem, we add a source and a sink node to the bidding zone's NTH topology. We add lines between the source node and all NTH with positive inner-BA net position and use the inner-BA net positions as capacities on these lines.

In the same way, we connect the NTHs with negative inner-BA net position to the sink node. All other lines correspond to intra-zonal nemo lines, and only the linear cost coefficients are applied. Given this input, a combinatorial minimum-cost maximum flow algorithm can be used to compute the flows on the NTH lines. The intra-BA SA flows are determined using the sum of the flows on the corresponding NTH lines.

Note that with this fallback, intra-BA inter-SA area NTH flows may not necessarily follow the same direction as the corresponding SA flow.

Annex C.4.Extended LTA inclusion

EUPHEMIA models the LTA inclusion by introducing some new variables and constraints, which are exhausted below. Also consider the LTA description on JAO, which is the one from the EUPHEMIA Lab R&D programme with NEMOs and TSOs, and which was the precursor this implementation. It is mostly identical with some minor changes in the exact modelling:

Input data

$PTDF_{z,p,r}: PTDF\ coefficient\ for\ bidding\ zone\ z, matrix\ p, row\ r.$

$RAM_{p,r}: remaining\ available\ margin\ on\ row\ r\ of\ the\ PTDF\ matrix\ p$

$LTA_{z,z',h}: LTA\ capacity$

Variables

Variable	Description	Range
$NET\_EXPORT\_FB_{z,t}$	net export of bidding zone z,period t flowing on the meshed FB network	Real
$NET\_EXPORT\_LTA_{z,t}$	Net LTA export of bidding zone z at period t flowing on the meshed FB network	Real
α_p:	the level of usage of the FB virgin domain for PTDF matrix p	[0..1]
$FB\_FLOW_{z,z',t}$	Flow on the virtual PTDF line from bidding zone z to bidding zone z' at period t **Note**: this variable is created at the time resolution of the balancing area, even if it is finer than the corresponding line.	Non-negative
$LTA\_FLOW_{z,z',t}$	LTA flow on the virtual PTDF line from bidding zone z to bidding zone z' at period t **Note**: this variable is created at the time resolution of the balancing area, even if it is finer than the corresponding line.	Non-negative

Constraints

Constraint	Description
$PTDF_{p,r}$: $\sum_{z \in balancing(p)} PTDF_{z,p,r} NET\_EXPORT\_FB_{z,t(p)} \leq \alpha_p RAM_{p,r}$	Where balancing(p) and t(p) are the balancing area and the period for which p is defined. These are the PTDF constraints of the virtual flow-based model.
$DEVIATION\_BALANCE_{a,t}$: $\sum_{z \in a} NET\_EXPORT\_FB_{z,t} + NET\_EXPORT\_LTA_{z,t} = 0$	This constraint states that the balancing area must be in balance with regards to the net exports on the FB network.

$MARKET\_EXPORT_{z,t}$: $NET_{POSITION}{}_{z,t} = res(z)\left(\sum_{l:from(l)=z}\left(FLOW_{l,t,up} - \left(1-loss_{l,t,down}\right)FLOW_{l,t,down}\right) + \sum_{l:to(l)=z}\left(FLOW_{l,t,down} - \left(1-loss_{l,t,up}\right)FLOW_{l,t,up}\right) + \sum_{t'\in sub(t,PTDF)}\frac{1}{\lvert sub(t,PTDF)\rvert}NET_{EXPORT_{FB}}{}_{z,t} + \sum_{t'\in sub(t,PTDF)}\frac{1}{\lvert sub(t,PTDF)\rvert}NET_{EXPORT_{LTA}}{}_{z,t}\right)$	The export constraint defines the net position of a bidding zone z in terms of its net export on the meshed network and the flows on the ATC lines connected to it according to their direction. Here $FLOW_{l,t,dir}$ refer to the flow at the parent period of t of the appropriate time resolution $\forall z, \forall t \in T(z)$
$CLEARING_{z,t}$: The clearing constraint associates the net position of the bidding zone to the matched orders. This constraint is not unique to LTA inclusion. It is how the net position in the MARKET_EXPORT constraint is associated to the matched orders	$\forall z, \forall t \in T(z)$
$LTA\_CAP_{z,z',t}$: $LTA\_FLOW_{z,z',t} \leq \left(1-\alpha_p\right)LTA_{z,z',t}$	This constraint states that the minimum and maximum LTA in a PTDF line must be respected $\forall (z,z') \in L, \forall t \in T(l)$ Where L is the set of LTA lines
$LTA\_DEVIATION_{z,t}$: $NET\_EXPORT\_LTA_{z,t} = \sum_{z'}\left(FLOW\_LTA_{z,z',t} - FLOW\_LTA_{z',z,t}\right)$	This constraint imposes the balance in the intuitive LTA sub-model. It decomposes the net position in terms of the virtual flows. Recall the time resolutions used are all the one from the balancing area.

$FLOW\_EQ_{z,z',t,t'}$: $FLOW_{z,z',t} = FLOW_{FB}{}_{z,z',t'} + FLOW_{LTA}{}_{z',z,t'}$	For lines between z and z′ where the time resolution of the line is coarser than that of the balancing area this constraint maps all line periods t to its underlying sub-periods t′ (e.g. a 60′ line has 4 15′ sub-periods for a 15′ balancing area). This constraint thus ensures a coarser line has the same flow value in all underlying sub periods. Per sub-period we do allow a different usage of the

	virgin FB domain or LTA domains. *Example*: If for an hourly line in a 15′ balancing area the flow will be 100MW for hour 1, this may be decomposed for the 4 underlying quarter hours: Q1: FLOW_FB = 100; FLOW_LTA = 0; Q2: FLOW_FB = 0; FLOW_LTA = 100; Q3: FLOW_FB = 50; FLOW_LTA = 50; Q4: FLOW_FB = 33; FLOW_LTA = 67; If this would allow for a more optimal allocation, benefiting the 15′ bidding zones inside this balancing area.

Annex C.5. Missing and Extra Money Management mechanism

The acceptance of curve order depends on the market price of its bidding zone. Missing Money Management mechanism requires that an order at a coarser time resolution that the one of its bidding zone can be paradoxically rejected.

Combination of different period orders can lead to price artifacts, including extreme prices far outside the price bounds. As multiple time resolutions have to be considered, simply capping the prices to the market bounds could lead to payment imbalance, and therefore missing or extra money. This should be infrequent but Euphemia is prepared to tackle these situations. To avoid such situations it was decided to only consider the prices at the smallest time resolution for capping, and recompute the capped prices at coarser time resolutions as the average of the prices of their sub-periods. While this prevents any possible missing or extra-money at nemo hub level, it causes other issues, such as paradoxically accepted curve orders.

We cannot at the same time have:

- an average rule valid on the capped price;
- avoidance of paradoxically accepted curve orders;
- avoidance of paradoxically rejected curve orders.

The 15′ market design does not allow paradoxically accepting any order types. Instead we allow curve orders of coarser time resolutions than that of the bidding zone, to be paradoxically rejected in specific situations.

Extra or missing money is avoided by an Euphemia mechanism ensuring that in any final output the Average Rule is strictly enforced. This however comes at the cost of having paradoxically rejected orders at the coarser time resolutions while prices are kept within the price bounds. Curve orders can be paradoxically rejected if the price needs to be capped in one of the sub-periods of the curve order's period. This means that curve orders defined at the time resolution of their bidding zone can under no circumstances be paradoxically rejected.

Mind that this mechanism is a heuristic: it may happen that coarser time resolution curve orders are paradoxically rejected, even if the published prices are all within the minimum and maximum clearing prices.

Average rule:

An important relation that can be obtained from the dual is the *Average Rule*. This rule states that the price at a parent period must be the average of the prices of its child periods. This rule must be respected even when the prices must be capped.

Simple example:

Notation here is *power@price*.

30'	C1: -10@p_{cl}	C2: -20@5
60'	C3: +15@7	

In this example 10MW will try to be matched. Step orders c3 and c2 are thus at-the-money and define the prices on their periods (p_{cl}). Step order c1 is fully accepted and a price p must be defined for this period. The average rule states

$$7 = \frac{p + 5}{2}$$

Which implies a price *p*= 9€/MWh. If p_{cl}≥ *p,* c1 is in-the-money and the prices are valid. If p_{cl}< *p,* the price is too high for c1 and nothing can therefore be matched.

Note that average rule ensures that there is no extra or missing money: c3 receives 10 * 7€, c2 pays 10/2 *5 €. c1 must thus pay 45€, which can be achieved by setting the price at 9€.

Example with extra-money:

Market prices must be between -550€/MWh and 3000€/MWh.

Considering the following orders:

30'	C1: -10@0	C2: +5@0
60'	C3: -10@2000	

c1 and c3 are curtailed, and will fix prices for their periods. The price *p* for c2 is still to be determined. According to the average rule:

$$2000 = \frac{p + 0}{2}$$

The price should thus be equal to 4000. However, due to capping, it will be lowered to 3000 and there is some extra money.

For the case where this extra money materializes inside a bidding zone (i.e. all orders are within the same bidding zone) this is solution is discarded through a cut. If the orders are located in different bidding zones, and the extra money materializes a congestion income on an uncongested interconnector, the solution is considered valid.

Example with missing-money:
Market prices must be between -550€/MWh and 3000€/MWh.

Considering the following orders:

30'	C1: +100@1000	C2: +80@2000
60'	C3: +20@2500	

with a 30' block order -90@3000 on both half-hours.

c1 and c3 are curtailed. The average rule expects the price in the second half-hour to be of 4000€/MWh. This price has to be capped at 3000€/MWh, leading to missing-money. The hourly price has thus to be lowered to 2000€/MWh. This hourly price makes c3 paradoxically accepted, which is not acceptable. A cut will be generated to kill this order, and nothing will be matched.

Annex C.5. Indexes and Annotations

Sets

Z	Set of all bidding zones
T(z)	Set of periods under the time resolution of bidding zone z

Indices

z	Bidding zone
t	Period

s	Supply/Demand
c	Curve identified by z,*t,s*
o	Period Order identified by z,*t,s,o*
bo	Block Order
mo	Merit order
po	PUN order
co	Complex Order, where • complex curve is identified by z*,co,t* • complex suborder by z*,co,t,o*
sco	Scalable Complex Order, where • Scalable complex curve is identified by z,sco,t • Scalable complex suborder by z,sco,t,o
Sign(type(sco))	+1 for supply -1 for demand
l	(DC/ATC) Line
uu(convention: up=0 and down=1)	Up/Down direction
Res(o)	Returns the weight of the time resolution of order o needed to convert power to energy. I.e. 15/60 (or ¼) for a 15′ order; 30/60 (or ½) for a 30′ order and 60/60 (or 1) for a 60′ order.
ACCEPT [0;1]	Acceptance variables
p	Offered Price *(in €/MWh)*
q	Offered quantity (in MW)
MCP	*Market clearing price (in €/MWh)*