NEMO COMMITTEE

EUPHEMIA Public Description

Single Price Coupling Algorithm

12th October 2020

Copyright disclaimer

All NEMOs disclaim any responsibility and all liability for content errors, omissions, or infringing material.

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.

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, Czech Republic, Croatia, Denmark, Estonia, Finland, France, Germany, Hungary, Italy, Ireland, Latvia, Lithuania, Luxembourg, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden and UK. PCR is implemented in both the MRC region as well as the 4M Market Coupling (4M MC).

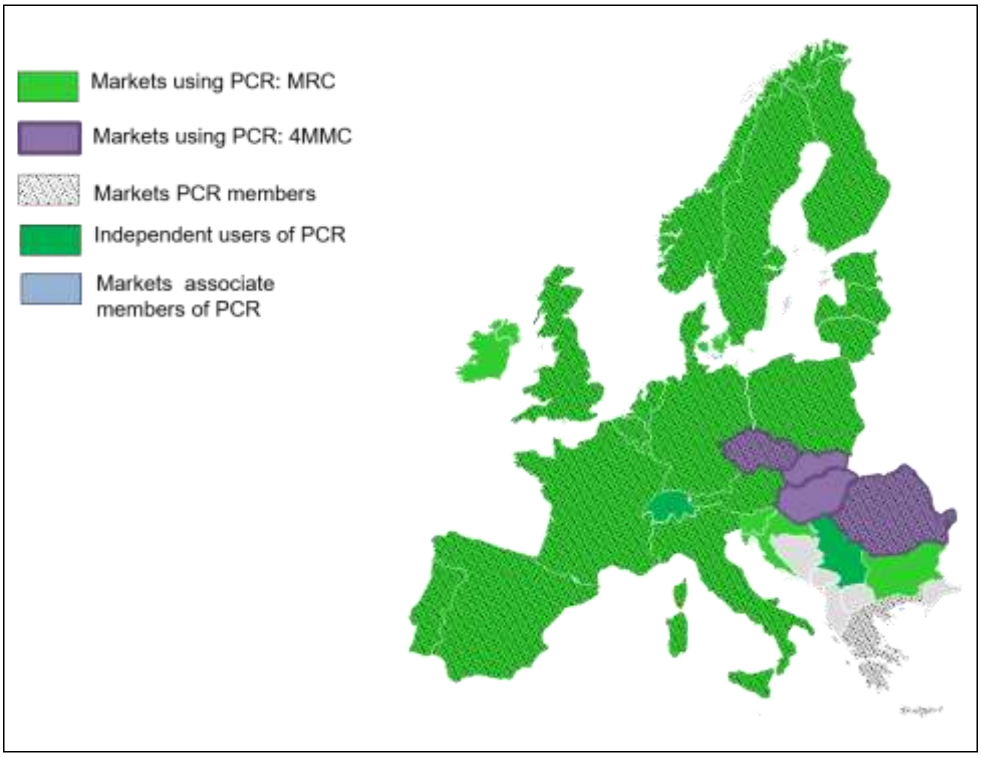


Figure 1 – PXs promoting PCR project

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 is progressively used to calculate energy allocation and electricity prices across Europe, maximizing the overall welfare 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.

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 *social welfare* (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 welfare. 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, no the 25th of February 2015, GME was successfully coupled. Recently, 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.

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. 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.

4.1.1. Net position ramping (hourly and daily)

The algorithm supports the limitation on the variations of the *net position* from one hour to the next. There are two ramping requirements that can be imposed on the *net position*.

- Hourly *net position* ramping: this is a limit on the variation of the *net position* of a *bidding zone* from one hour 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 hour 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 hour for the previous (delivery) day, overnight ramping can be taken into account.

4.2. 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 4.2.1) of the line.

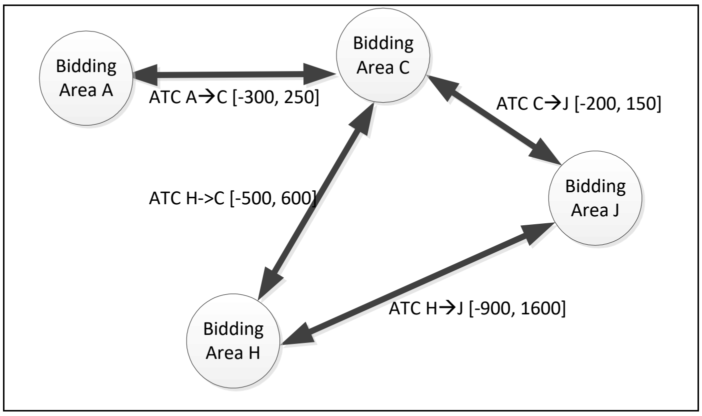


Figure 2 – *Bidding zones* connected in ATC model

Additional restrictions may apply to the interconnectors:

- The flow through a line can be subject to losses (Section 4.2.2)
- The flow through a line can be subject to tariffs (Section 4.2.3)
- The flow variation between two consecutive hours can be restricted by an hourly flow ramping limit (Sections 4.2.4 and 4.2.5)

4.2.1. 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).

4.2.2. 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).

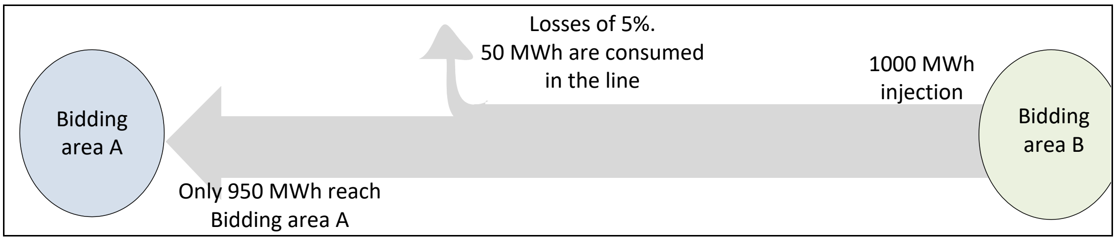

Losses of 5%.
50 MWh are consumed in the line
1000 MWh injection
Bidding area A
Bidding area B
Only 950 MWh reach Bidding area A

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

4.2.3. 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.

4.2.4. Hourly Flow Ramping Limit on Individual Lines

The hourly 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 hour to the next (Figure 4). The ramping limit constrains the flow that can pass through the line in hour h depending on the flow that is passing in the previous hour h-1.

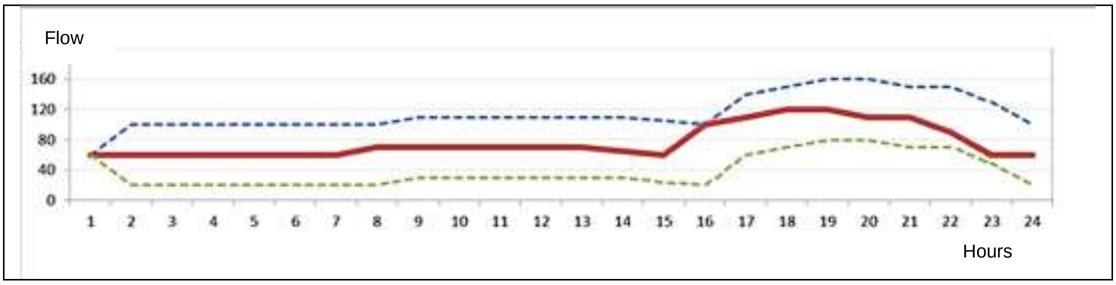


Figure 4 – Effect of the hourly flow ramping limit. The flow stays in the allowed band between hours.

The ramping limit is defined by: The maximum increment of flow from hour h-1 to hour h (called ramping-up), and the maximum decrement of flow from hour h-1 to hour h (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 hour of the previous day.

4.2.5. Constraints on Line Sets

4.2.6. Hourly 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 and the latter between areas A and C. If we set the hourly 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 hour to the next.

4.2.7. 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.

4.2.8. 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.

4.2.9. External constraint

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

In EUPHEMIA the $\text{NET\_POSITION}_{m,h}$ 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$_{m,h}$ variable, to limit the global import/export of each bidding zone at each period:

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

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

4.3. 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.

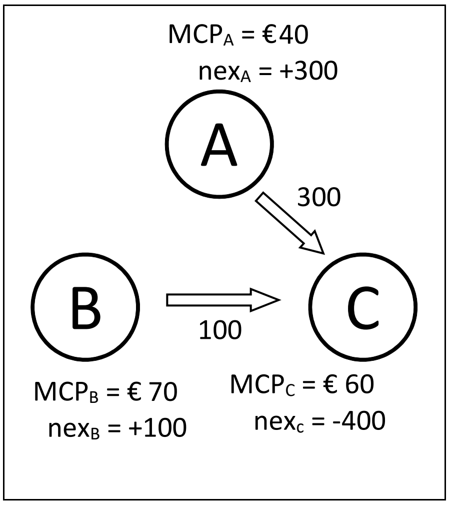


Figure 5 – Example of *net position*s decompositions into flows

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.

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. Welfare maximization can therefore lead to these non-intuitive situations.

EUPHEMIA integrates a mechanism to suppress these non-intuitive exchanges. This mechanism seeks "flows" between areas which match the *net position*s. Rather than imposing the PTDF constraints directly on the *net position*s, in intuitive mode they are applied to these "flows". So far the two models are fully equivalent. However in case a PTDF constraint is detected that leads to a non-intuitive situation, all of its relieving effects are discarded: the impact of a "flow" from *i* to *j* actually is $PTDF_i - PTDF_j$, but is replaced by $max(PTDF_i\text{-}PTDF_j, 0)$.

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 welfare 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.

4.3.1. 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

4.4. Scheduling Area Topology

4.4.1. 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.

4.4.2. 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 6.8.5 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.

4.5. NEMO Trading Hub Topology

4.5.1. 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.

4.5.2. 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 6.8.5 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 Hourly Orders
- Complex Orders
    - MIC orders
    - Load Gradient orders
- Block Orders
    - Linked Block Orders
    - Exclusive Groups of Block Orders
    - Flexible Hourly Orders
- Merit Orders and PUN Orders.

5.1. Aggregated Hourly 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 each period of the day. 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*).

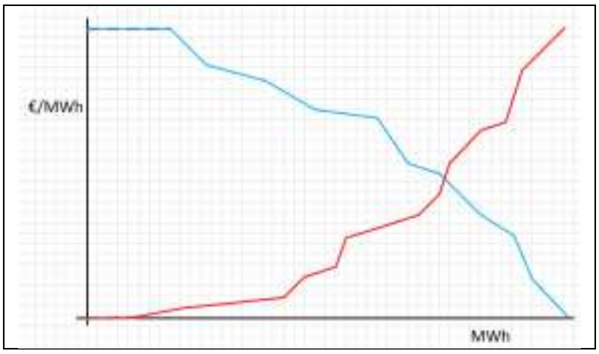


Figure 6 – Linear piecewise aggregated curve.

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

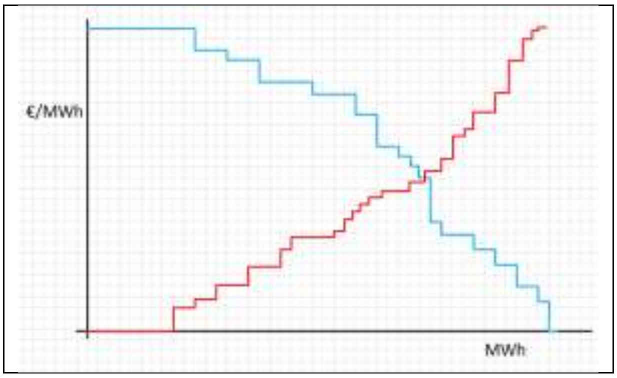


Figure 7 – Stepwise aggregated curve.

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

The following nomenclature is used when speaking about hourly orders[2] and market clearing prices:

- One demand (resp. supply) hourly order is said to be *in-the-money* when the market clearing price is lower (resp. higher) than the price of the hourly order.
- One demand or supply hourly order is said to be *at-the-money* when the price of the hourly order is equal to the market clearing price.
- One demand (resp. supply) hourly order is said to be *out-of-the-money* when the market clearing price is higher (resp. lower) than the price of the hourly order.
- For linear piecewise hourly 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 market clearing price is in the interval $[p_0, p_1]$).

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

- Any order in-the-money must be fully accepted.

[2] Whenever hourly orders are mentioned through this document, we are referring to the aggregated hourly orders that are the input of EUPHEMIA.

- Any order out-of-the money must be rejected.
- Orders at-the-money can be either accepted (fully or partially) or 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 hourly orders (which are referred to as hourly 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 hourly sub-orders as a whole.

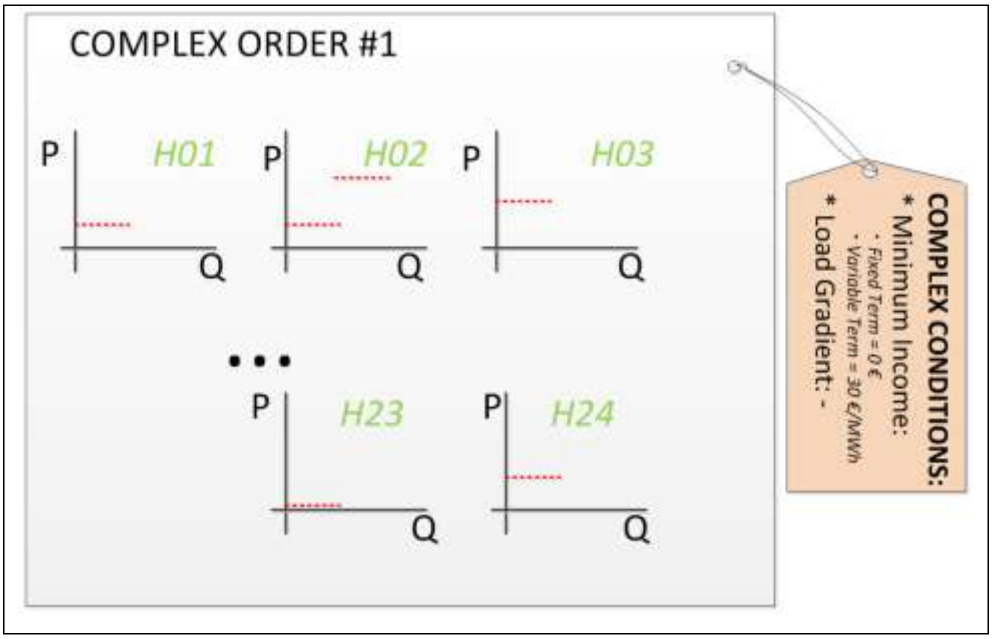


Figure 8 – A complex order is composed of a set of hourly sub-orders (in dotted line) associated with complex conditions

Complex conditions are of two types: Minimum Income Condition (with or without scheduled stop), and Load Gradient.

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 5.2.5

5.2.1. Minimum Income Condition (MIC)

Complex orders (with their set of hourly sub-orders) subject to Minimum Income Condition constraints are called MIC orders (or MICs).

Generally speaking, the Minimum Income economical 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 (representing the startup cost of a power plant) and a variable term multiplied by the total assigned energy (representing the operation cost per MWh of a power plant).

The Minimum Income Condition constraint is in short defined by:

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

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

- In case a MIC order is activated, each of the hourly sub-orders of the MIC behaves like any other hourly order, which means accepted if they are in-the-money and rejected if they are out-of-the-money, and can be either accepted (fully or partially) or rejected when at-the-money.
- In case a MIC order is deactivated, each of the hourly sub-orders of the MIC is fully rejected, even if it is in-the-money (with the exception of scheduled stop, see Section 5.2.2).

The final solution given by EUPHEMIA will not contain active MIC orders not fulfilling their Minimum Income Condition constraint (also known as paradoxically accepted MICs).

5.2.2. Scheduled Stop

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 hourly sub-orders. On the contrary, the first (i.e. the cheapest) hourly sub-order in the periods that contain scheduled stop (up to period 3) will not be rejected but will be treated as any hourly order.

5.2.3. Load Gradient

Complex orders (with their set of hourly 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 hourly sub-orders belonging to a Load Gradient order in one period is limited by the amount of energy that was matched by the hourly 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 hour of the previous day. If only one of these values is defined, the other value (i.e. empty) is considered as unconstrained.

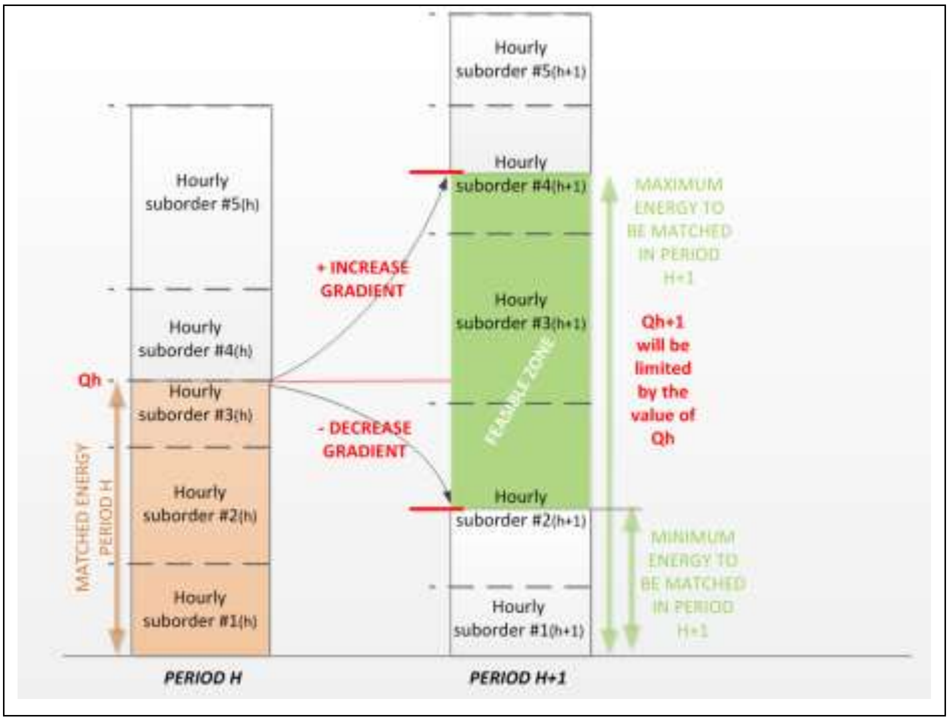


Figure 9 – A Load Gradient order. Effect produced by the amount that is matched in period (h) on period (h+1).

5.2.4. Complex orders combining Load Gradient and MIC

Complex orders (with their set of hourly sub-orders) can be subject to both Load Gradient and Minimum Income Condition (with or without scheduled stop).

5.2.5. 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 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;
    - Quantities;

For this case, economic criteria are insufficient to arbitrate: accepting one or the other will result in identical welfare. 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 behavior 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 hourly orders (which are referred to as hourly 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 hourly sub-orders as a whole.

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

- A set of stepwise hourly sub-orders (sell for scalable MIC orders; buy for scalable MP orders), one set per period;
- An economic condition, which represents the minimum income (respectively, the maximum payment) expected by order's owner defined by a fix term (FT) in Euros and the price of each sub-order in the set of hourly sub-orders in euros per accepted MWh.
- A minimum acceptance volume per hour.

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

- If the economic condition is not fulfilled, the scalable complex order must be rejected.
- If the economic condition is fulfilled, the scalable complex order can be accepted.

- If the 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 volume less than the minimum acceptance volume defined for all and each one of the periods.

The final solution given by EUPHEMIA will not contain active SCO orders not fulfilling their economic condition constraint (also known as paradoxically accepted SCOs).

SCO orders also support the scheduled stop condition. This condition only applies to deactivated scalable MIC orders and only in the periods declared as part of the scheduled stop interval by the scalable MIC order. In case in which a scalable MIC order is deactivated, the first hourly sub-order of the set of offers belonging to the deactivated scalable MIC order in a period declared part of the scheduled stop interval will remain activated and they will be accepted if they are in-the-money and could be accepted if they are at-the-money.

Finally, the sender of a SCO has the possibility to define load gradient constraints as it happened with the classical complex orders, described in section 5.2.3 Load Gradient

This load gradient condition limits the variation between the accepted volume of an order in a period and the accepted volume of the same order in the adjacent periods, according to an increase gradient and/or a decrease one and come with or without economic condition. Between two consecutive periods, the accepted volume of the SCO with load gradients cannot vary by more than the defined gradients.

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),
- number of periods,
- 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: Hours (3-7), hours (8-19) and hours (22-24)
- Volume: 80 MWh in the first interval, 220 MWh in the second one, and 40 MWh in the third one.

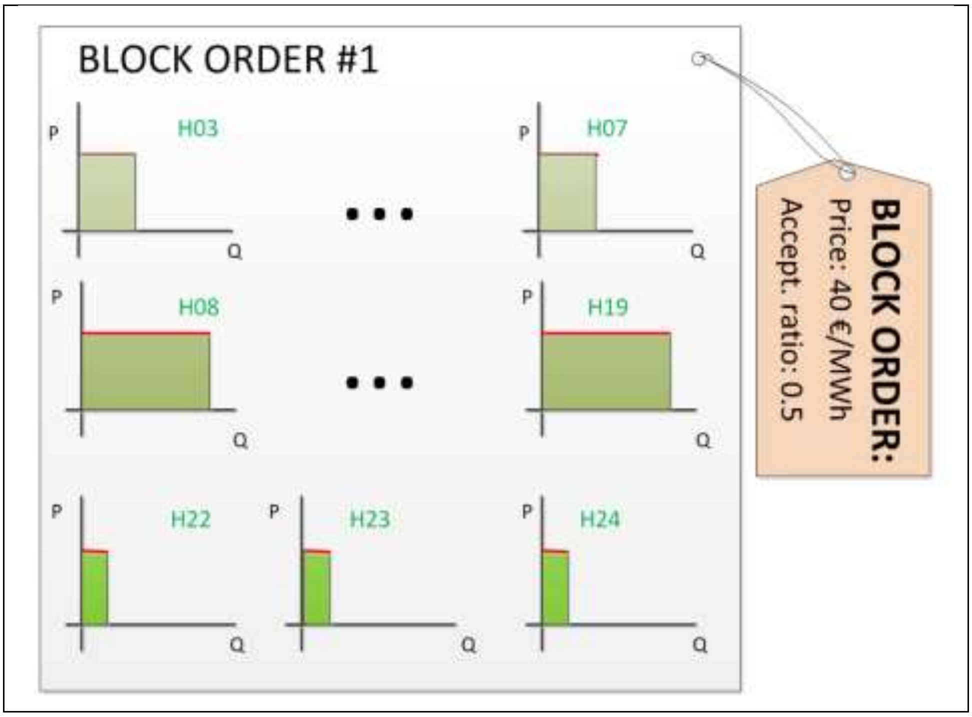


Figure 10 – Block order example

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;

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 hourly 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".

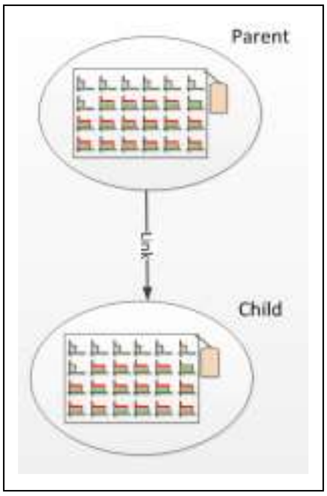


Figure 11 – Linked block orders

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 welfare 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 (*social welfare*, see Section 6.3).

5.4.3. Flexible Hourly Orders

A flexible "hourly" order is a block order with a fixed price limit, a fixed volume, minimum acceptance ratio of 1, with duration of 1 hour. The hour is not defined by the participant but will be determined by the algorithm (hence the name "flexible"). The hour in which the flexible hourly order is accepted, is calculated by the algorithm and determined by the optimization criterion (see Section 6.3)

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 welfare. 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.

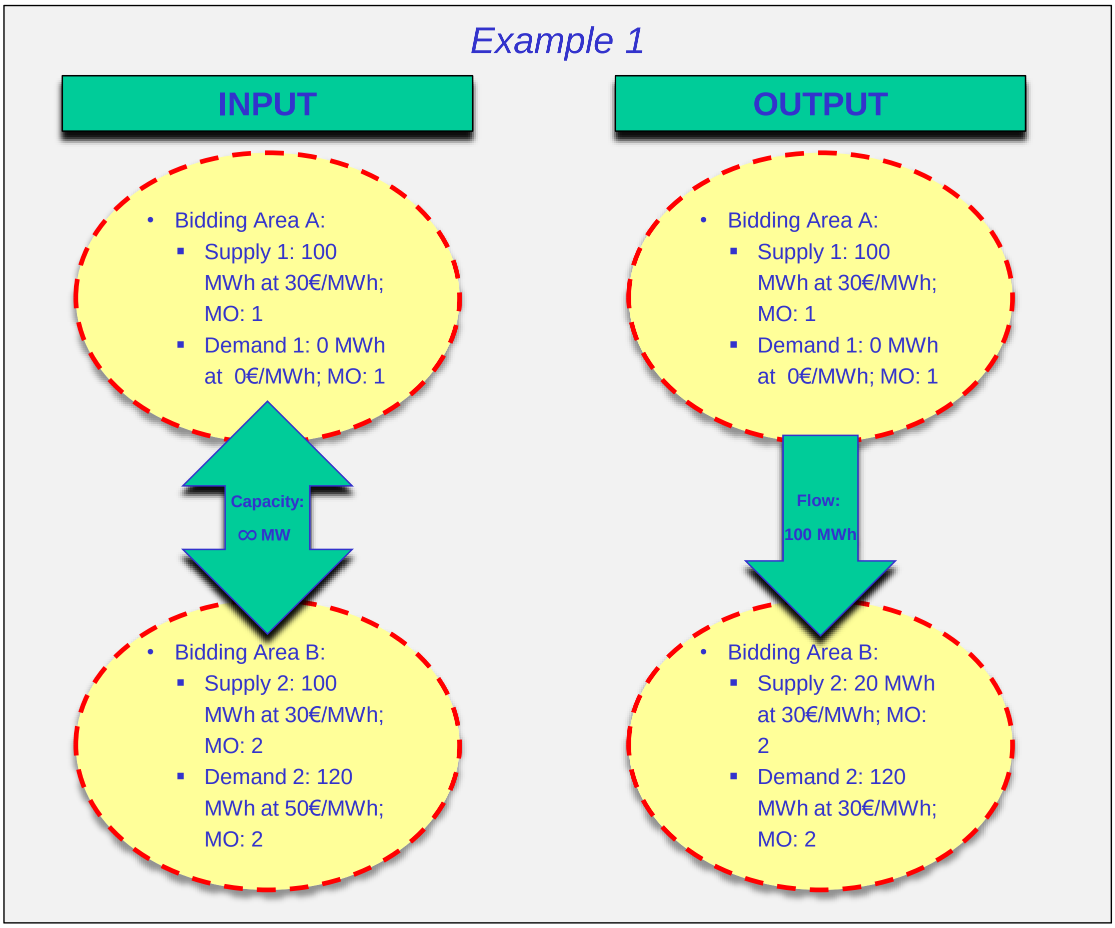
In order to be sure EUPHEMIA behaviour is repeatable, repeatable randomness is applied. This is managed by using the hashes that were compiled for each block (on the basis of the different parameters describing the blocks). These hashes will be used to settle ties, and should be sufficiently random to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

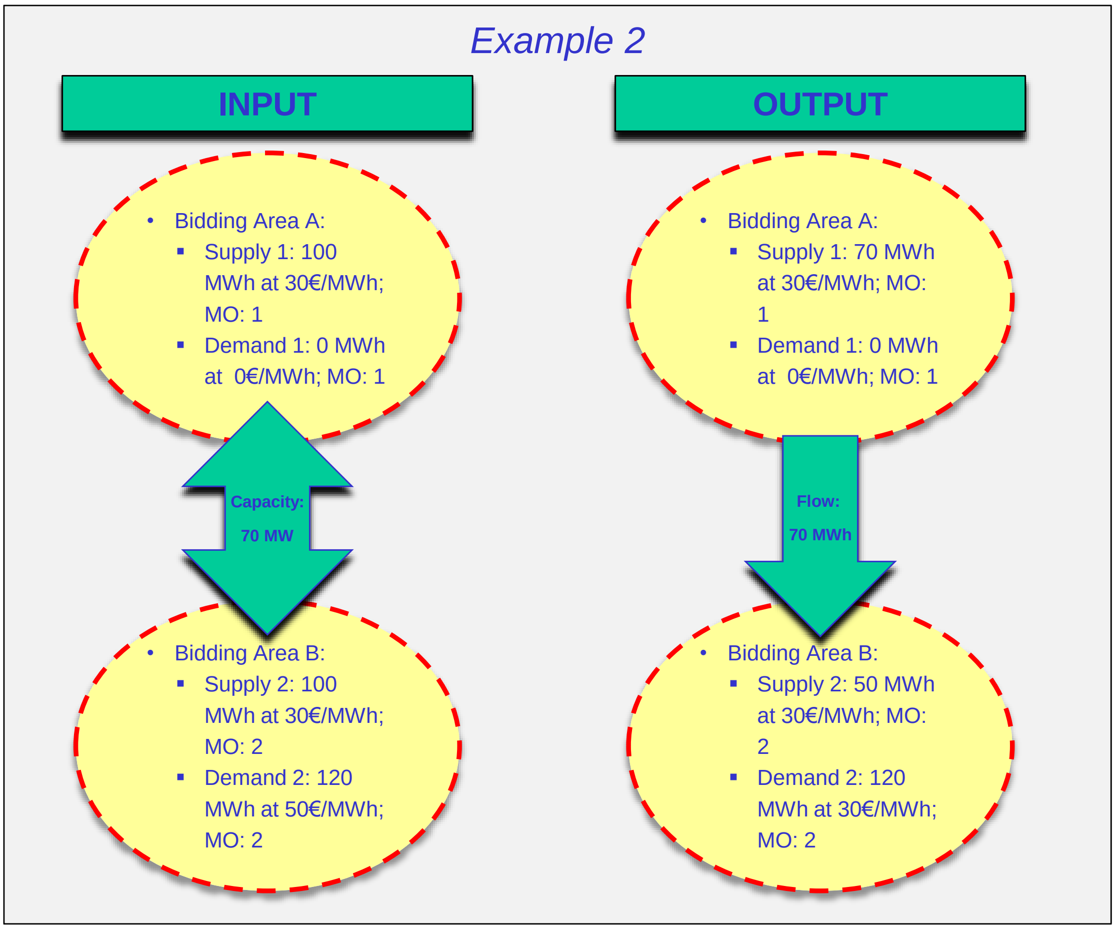

Example 2
INPUT
OUTPUT
Bidding Area A:
Supply 1: 100 MWh at 30€/MWh; MO: 1
Demand 1: 0 MWh at 0€/MWh; MO: 1
Capacity: 70 MW
Bidding Area B:
Supply 2: 100 MWh at 30€/MWh; MO: 2
Demand 2: 120 MWh at 50€/MWh; MO: 2
Bidding Area A:
Supply 1: 70 MWh at 30€/MWh; MO: 1
Demand 1: 0 MWh at 0€/MWh; MO: 1
Flow: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

Figure 12: 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 prices* (up to a defined tolerance named PUN imbalance[3]), 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

[3] 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.

- 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.

6. EUPHEMIA Algorithm

6.1. 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 (4.4.1 and 4.5.1), 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, 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.

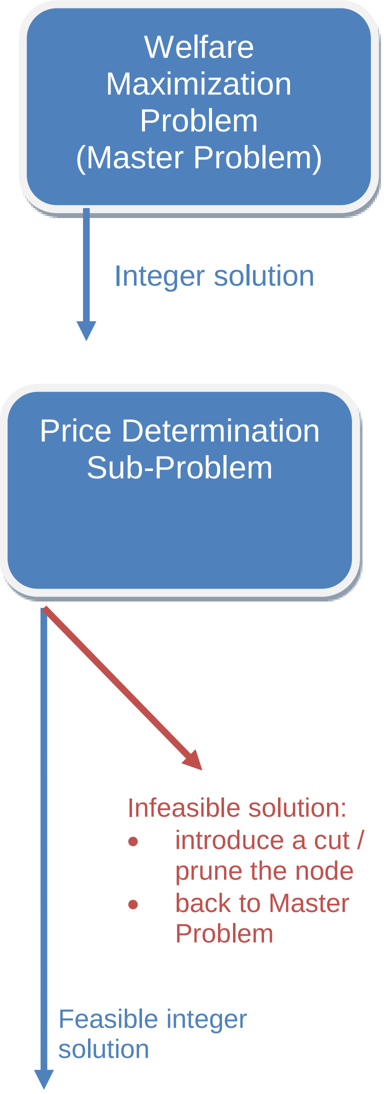
To retrieve the results at NTH level, EUPHEMIA also implements a disaggregation post-processing step, once solutions have been found.

6.2. 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 *social welfare*, *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.

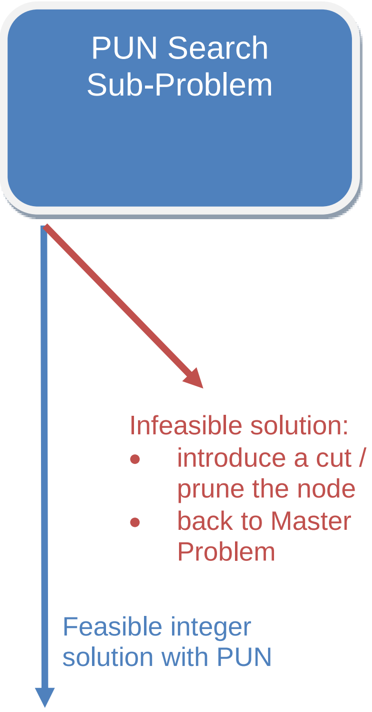
By ignoring the particular requirements of the block, complex, merit and PUN orders, the market coupling problem resolves into a much simpler problem which can be modeled as a Quadratic Program (QP) and solved using commercial off-the-shelf solvers. However, the presence of these orders renders the problem 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 modeling 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 welfare 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.

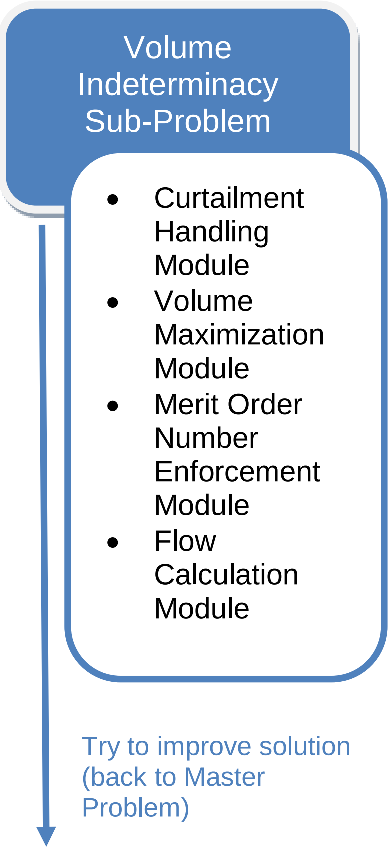

Welfare Maximization Problem (Master Problem)
Integer solution
Price Determination Sub-Problem
Infeasible solution:
introduce a cut / prune the node
back to Master Problem
Feasible integer solution

In the welfare maximization problem, EUPHEMIA searches among the set of solutions (solution space) for a good selection of block and MIC orders that maximizes the *social welfare*. 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 welfare maximization problem must be rejected. This is achieved by adding a cut to the welfare maximization problem that renders its current solution infeasible. Subsequently, EUPHEMIA resumes the welfare maximization problem searching for a new integer solution for the problem.


PUN Search Sub-Problem
Infeasible solution:
introduce a cut / prune the node
back to Master Problem
Feasible integer solution with PUN

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 welfare maximization problem that renders the current solution infeasible. Otherwise, EUPHEMIA proceeds with the lifting of volume indeterminacies.


Volume Indeterminacy Sub-Problem
Curtailment Handling Module
Volume Maximization Module
Merit Order Number Enforcement Module
Flow Calculation Module
Try to improve solution (back to Master Problem)

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 hourly volumes, *net positions*, and bidding zone line flows that are coherent with these prices and that yield the same welfare. 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 6.8.5.

6.3. Welfare Maximization Problem (Master Problem)

As mentioned previously, the objective of this problem is to maximize the *social welfare*, *i.e.* the total market value of the Day-Ahead auction. The *social welfare* 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 0.

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

- The acceptance criteria for aggregated hourly demand and supply curves and merit orders
- The fill-or-kill requirement of block orders

1. The scheduled stop, load gradient, and minimum income condition of 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 hourly and daily 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 hourly orders while, the PUN orders are just ignored. The main difficulty of the welfare 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 modeled 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 welfare 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.

6.3.1. 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 *social welfare*, it always gives an upper bound on attainable *social welfare*. 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.

6.3.2. 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).

6.3.3. 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 6.4). 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 *social welfares*. 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 welfare maximization problem is created where additional constraints are added to the welfare maximization problem that renders the previous integer solution infeasible (see section 6.3.4).

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

6.3.4. 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 *social welfare* 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 welfare maximization problem must be rejected. To do so, specific local cuts are added to the welfare 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.

6.4. 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 hourly 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 welfare 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 fulfill 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:

2. 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 welfare maximization problem.
3. 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.

6.4.1. Branch-and-Cut Example

Here is a small example of the execution of the Branch-and-Cut algorithm (Figure 13).

At the start of the algorithm, we do not have an incumbent solution. EUPHEMIA first solves the relaxed welfare maximization problem where all the integrality constraints have been relaxed (Instance A). Let us assume that the solution of this problem has a *social welfare* 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 *social welfare* 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 welfare* 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 *social welfare* is a lower bound on any achievable welfare 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 *social welfare* equal to 3440. As the obtained *social welfare* 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 welfare 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 *social welfare* is equal to 3440.

Now, let us assume that when solving the instance D of the problem, we get a solution with a *social welfare* 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 *social welfare* equal to 3200. This *social welfare* 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 *social welfare* is now equal to 3200.

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

Figure 13 shows the search tree associated with our example.

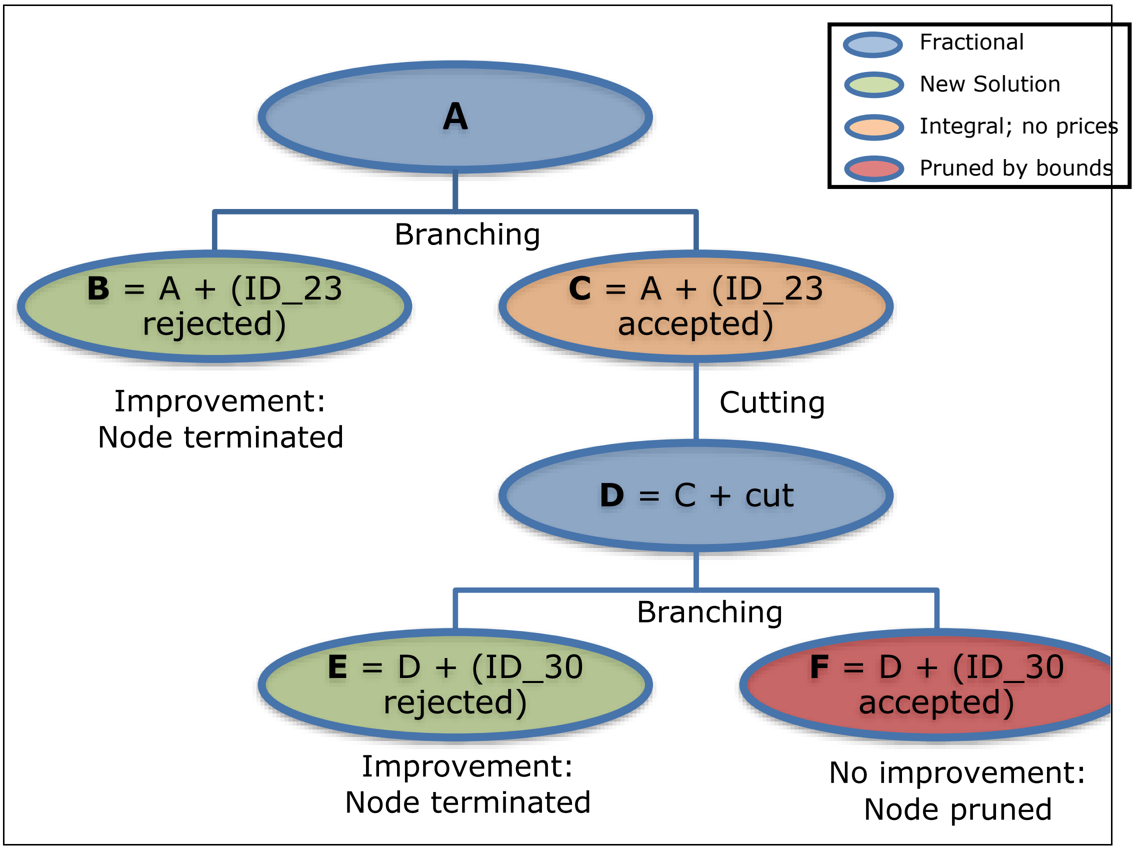
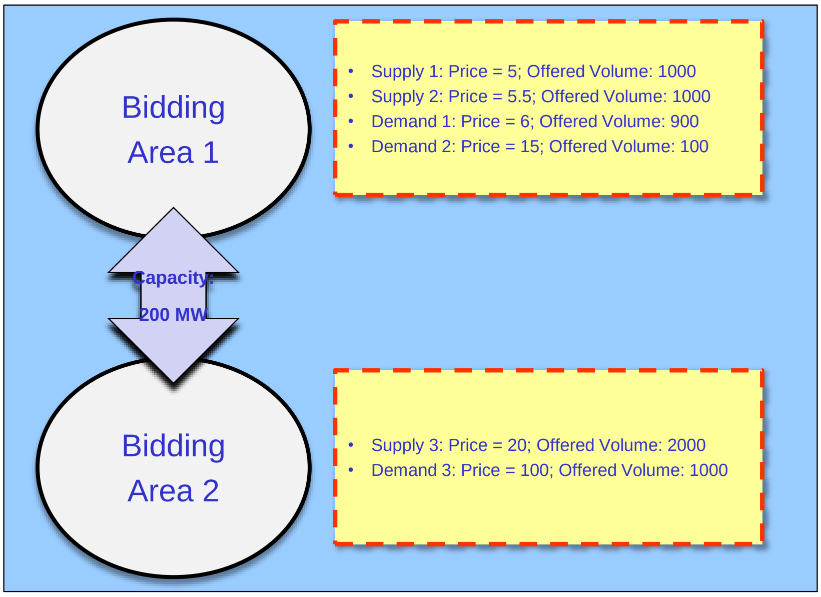


Figure 13 - Branch-and-Cut example

6.5. PUN Search Sub-problem

In order to avoid *paradoxically accepted* PUN orders, PUN (see Section 6.5) 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:



- Supply 1: Price = 5; Offered Volume: 1000
- Supply 2: Price = 5.5; Offered Volume: 1000
- Demand 1: Price = 6; Offered Volume: 900
- Demand 2: Price = 15; Offered Volume: 100

- Supply 3: Price = 20; Offered Volume: 2000
- Demand 3: Price = 100; Offered Volume: 1000

Figure 14 – PUN acceptance

If in Figure 15, 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:

$$[(1000 * 5.5) + (1000 * 20)] / 2000 = 12.75 \text{ €/MWh}.$$

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;
- Welfare: (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 16), 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.

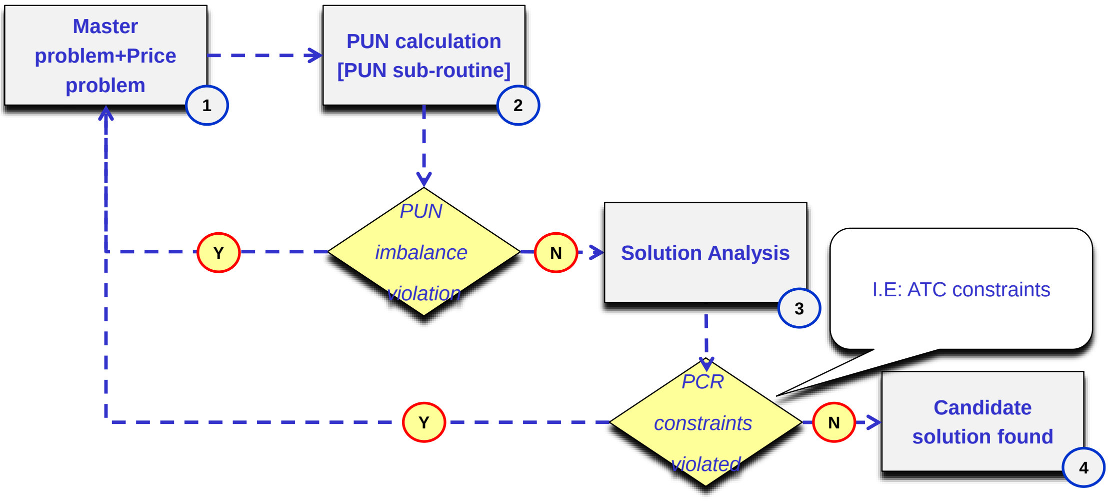


Figure 16 – PUN Search Sub-problem process

The PUN search is essentially an hourly sub-problem where the requirements are defined on an hourly 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.

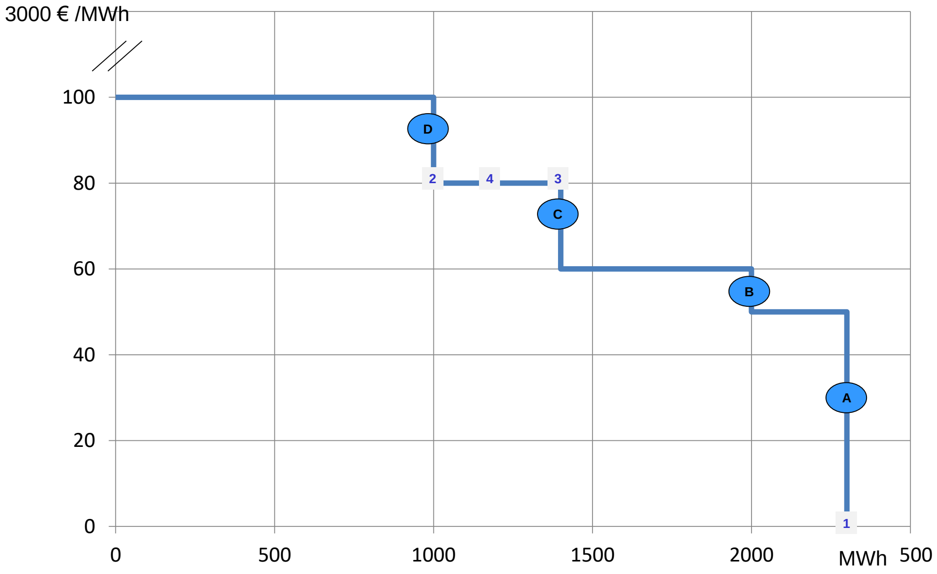


Figure 17 – PUN hourly curve

EUPHEMIA starts by calculating the PUN imbalance associated with the maximum accepted PUN volume (negative imbalance expected[4]; point 1 in Figure 17). If the PUN imbalance associated with the maximum PUN doesn’t violate PUN imbalance tolerance, a candidate solution is found.

[4] 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 17, 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 17, 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 17, positive PUN Imbalance in segment D; and negative PUN Imbalance in segment C). In this case, EUPHEMIA fixes the price (i.e. in Figure 17, 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 17, 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 welfare maximization problem that renders its current solution infeasible. Otherwise, EUPHEMIA proceeds with the PRMIC reinsertion.

6.6. PRMIC reinsertion

Finally, if the PUN sub-problem is successful, the solution returned by EUPHEMIA should be made free of any false paradoxically rejected complex MIC order (PRMIC). Thus, once the market clearing prices have been found, EUPHEMIA proceeds with an iterative procedure aiming to verify that all the rejected complex MIC 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 complex MIC 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 *social welfare* 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 *social welfare* is reduced, the complex MIC 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 welfare 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.

6.7. 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 welfare.

As soon as a solution has been stored, a local search algorithm tries to find neighbor 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 welfare 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.

6.8. 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 welfare definition;
- Volume maximization
- Merit order indeterminacy
- Flow indeterminacy

6.8.1. 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*. 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*. Hourly supply orders at the minimum price of this range and hourly 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 hourly orders* with hourly 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 an hourly 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 18 where the dotted line indicates the minimum of *price-taking supply quantity* to be accepted).

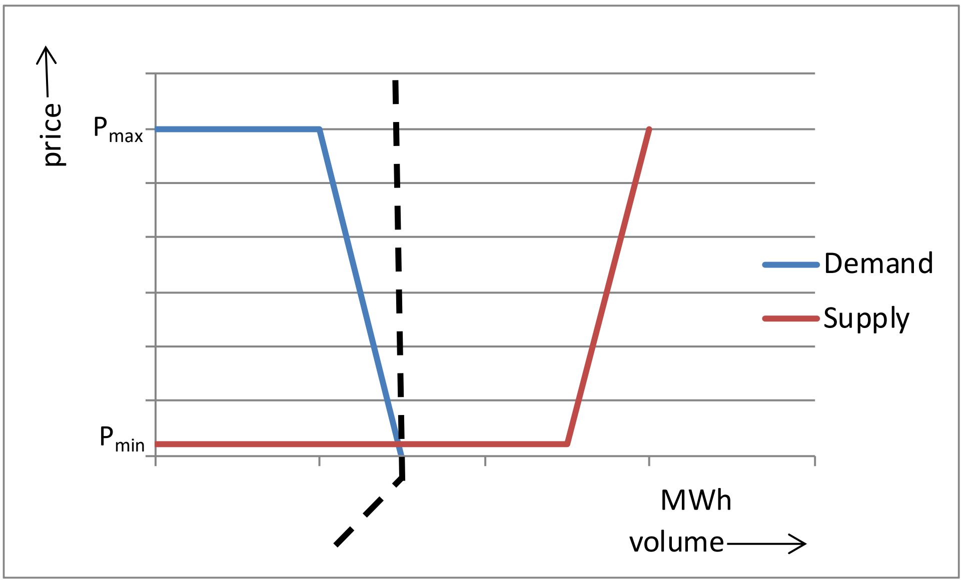


Figure 18 – Dotted line indicates the minimum of (*price-taking*) supply volume to be accepted

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 welfare maximization problem.

6.8.2. 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 areas 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[5]

The objective function of the master problem is to maximize welfare. 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 welfare 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 welfare. 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 welfare. 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 welfare 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 welfare 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

[5] 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).

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:

$$\begin{aligned}\min \quad & \sum_{h} \sum_{\substack{m: \\ m \in C_{h,Supply}}} \sum_{\substack{o: \\ market(o)=m \\ \mathrm{Po}=\mathrm{p}_{\min,m}}} |\mathrm{q}_o| \left(1 - ACCEPT_o\right)^2 \\ & + \sum_{h} \sum_{\substack{m: \\ m \in C_{h,Demand}}} \sum_{\substack{o: \\ market(o)=m \\ \mathrm{Po}=\mathrm{p}_{\max,m}}} |\mathrm{q}_o| \left(1 - ACCEPT_o\right)^2\end{aligned}$$

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.

6.8.3. Maximizing Accepted Volumes

In this step, the algorithm maximizes the accepted volume.

All hourly orders, complex hourly 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.

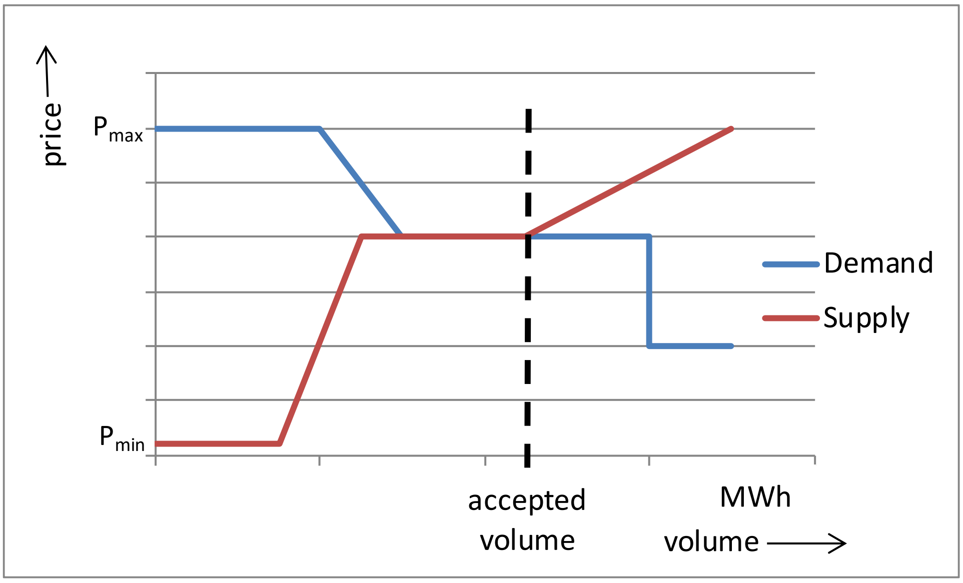


Figure 19 – The accepted volume is maximized

6.8.4. Merit order enforcement

This step enforces merit order numbers of the hourly orders if applicable. The acceptance of hourly 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.

6.8.5. 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 0 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.

6.9. Multi-threading approach

In order to improve the quality of the solution (i.e. the limit the number of PRBs), to increase welfare 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:

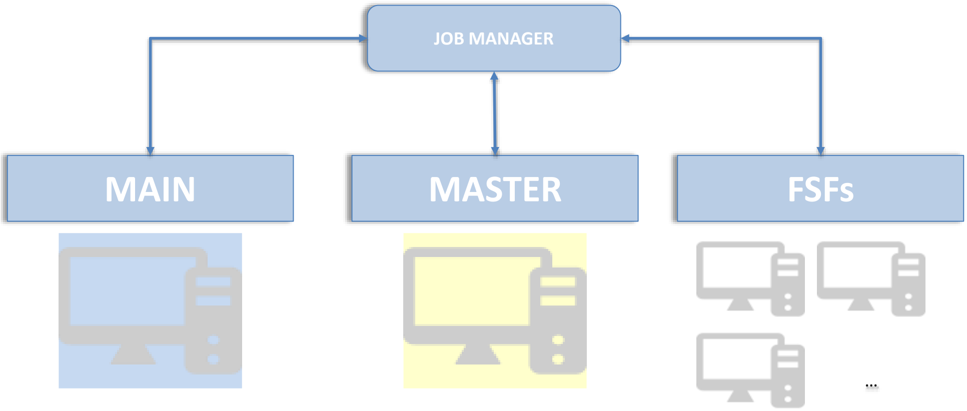


Figure 20 – EUPHEMIA 10 architecture

- **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 the figure below:

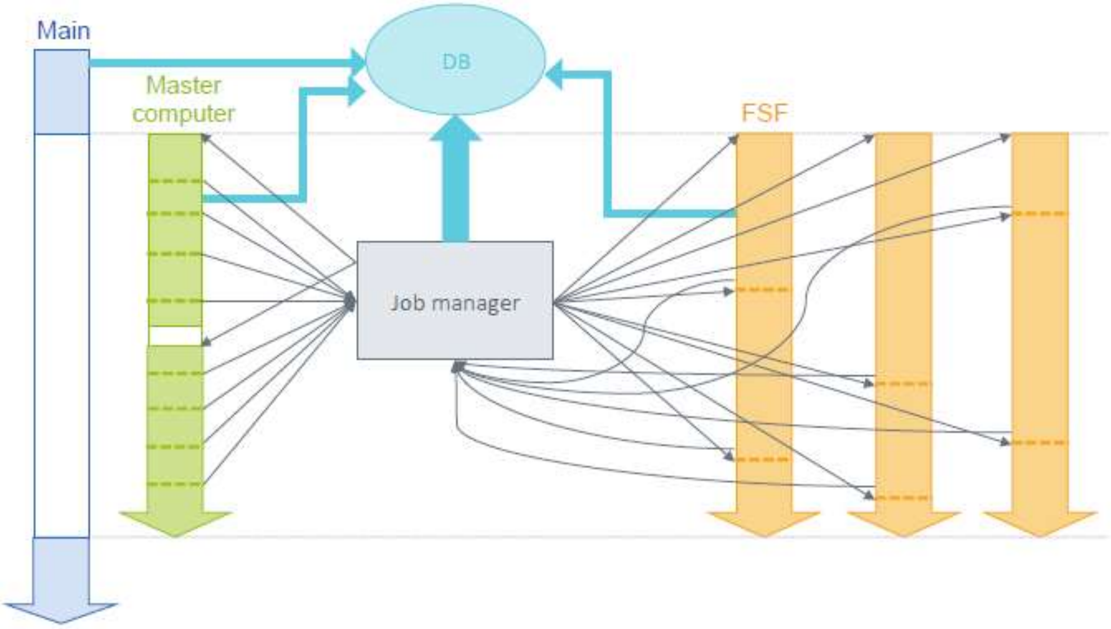


Figure 21 – Interaction actors

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.

6.9.1. 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.

6.9.2. 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.

6.9.3. 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).

6.9.4. 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.

6.9.5. 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.

6.10. 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:

o 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.

o 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.

7. Additional Requirements

7.1. 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.

7.2. Properties of the solution

During the execution of EUPHEMIA, several feasible solutions can be found. However, only the solution with the highest welfare value (complying to all network and market requirements) found before the stopping criterion of the algorithm is met is reported as the final solution.

It should be noted that for difficult instances some heuristics[6] are used by EUPHEMIA in its execution. Thus, it cannot be expected that the "optimal" solution is found in all cases.

7.3. Transparency

EUPHEMIA produces feasible solutions and chooses the best one according to the agreed criterion (welfare-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.

[6] 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)).

Annex A. Glossary

Item	Description
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. For blocks this notion is generalized by considering the volume weighted average price.
Bidding zone	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).
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.
Interconnector	a physical connection between two hubs
In-the-money	A supply (demand) order is considered in-the-money if its price is smaller (greater) than the market clearing price. For blocks this notion is generalized by considering the volume weighted average price.
Line	an abstract representation that connects two bidding zones;
Market Clearing Price (MCP)	A common reference price for the whole Market area, when not considering transmission constraints.
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. 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
Price-taking orders	Price taking orders (PTOs) are hourly buy (resp. sell) orders at the maximum (resp. minimum) price. PTOs are not block orders.
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).
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 hourly orders made by PUN orders (“consumption purchase hourly orders”).
Scheduling Area (SA)	Delivery area within a bidding zone, i.e. typically an area under the control of a TSO
Social welfare	The Social Welfare is defined as the sum of the Consumer Surplus, the Producer Surplus, and the Congestion Rent.
TSO	Transmission System Operator

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.

Intuitive Flow Based

The implementation of intuitive flows (exchanges from low to high price) under flow based capacity constraints, is managed by heuristics, losing some guarantee of optimality. These heuristics will be outlined in the following text.

As explained in section 4.3, when EUPHEMIA fails to find a breakdown of the net positions into intuitive "flows", it substitutes the constraints by a variant that discards relieving effects.

We will illustrate by a simple example (taken from http://www.jao.eu/DynamicContent/DownloadFile?url=ceq0h4gi.l5n.pdf&filename=Annex+16_12+Intuitiveness+Report.pdf&openInBrowser=false).

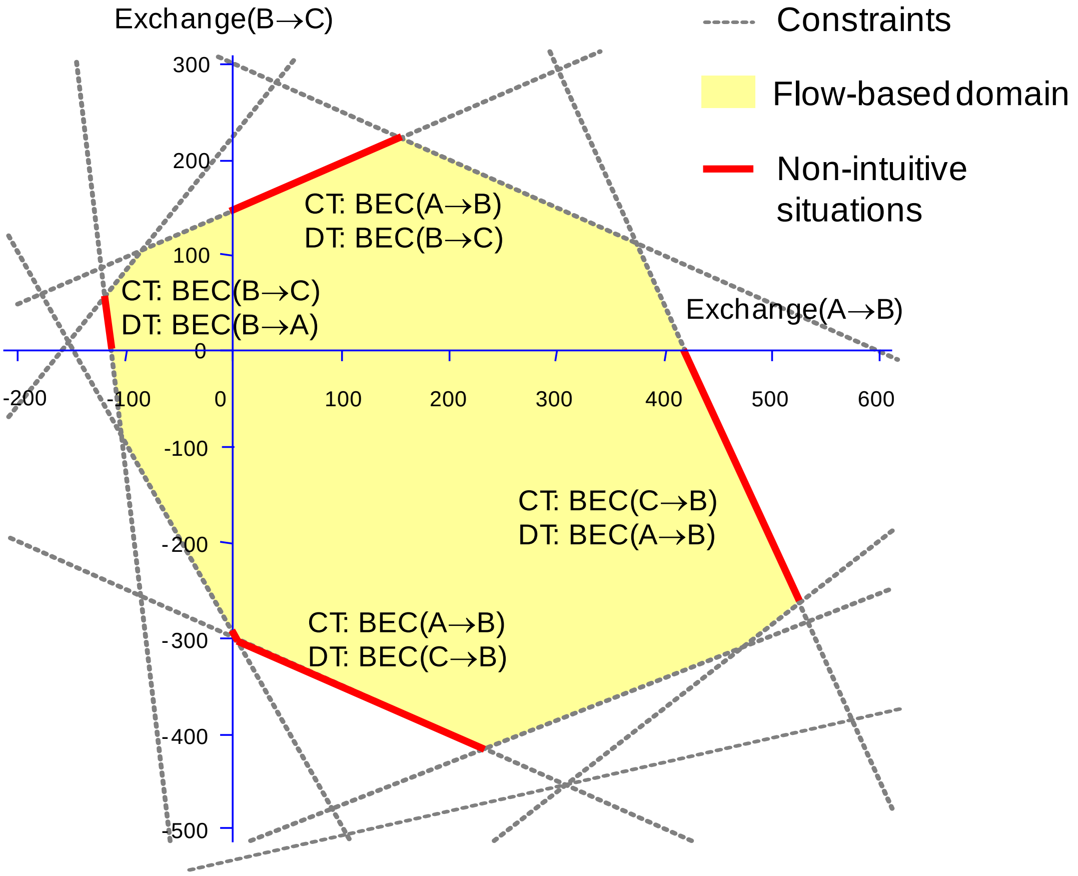
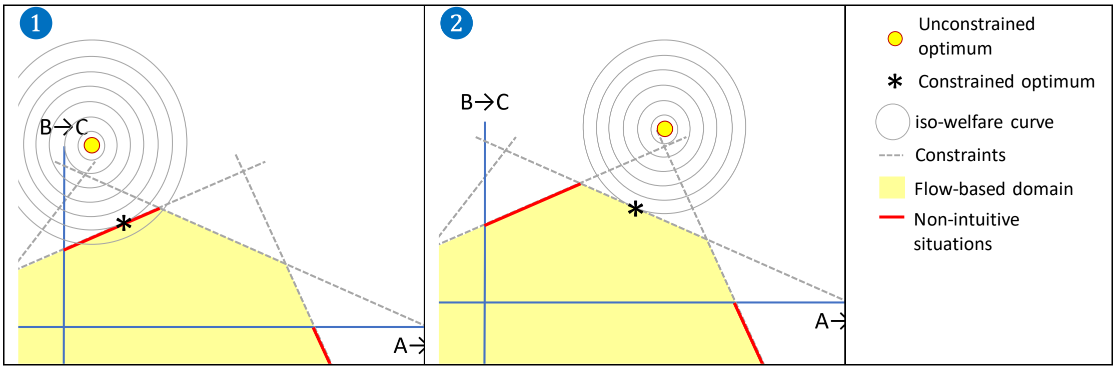


Figure 20 Illustration of non-intuitive situation, with CT: countertrade; DT: direct trade

Consider the diagram of Figure 20: on the axes we have bilateral exchange calculations (or BEC, or "flows" in our terminology) between pairs of bidding zones (A-B and B-C respectively). The yellow shaded area reflects the FB domain, market off by dotted lines, which reflect the FB constraints for each of the critical network elements.

Realise that if EUPHEMIA finds a welfare maximising solution within the FB domain, this implies none of the FB constraints is binding. I.e. all prices should fully converge, as there is no congestion.

More interesting are the situations where we find a solution on one of the edges of the domain: i.e. there is a congestion, so we can expect clearing prices to differ. We will now provide a qualitative reasoning when we can expect solutions to be intuitive, and when we can expect them to be non-intuitive. We will focus on the first quadrant of Figure 20, but similar reasoning can be applied to any quadrant:



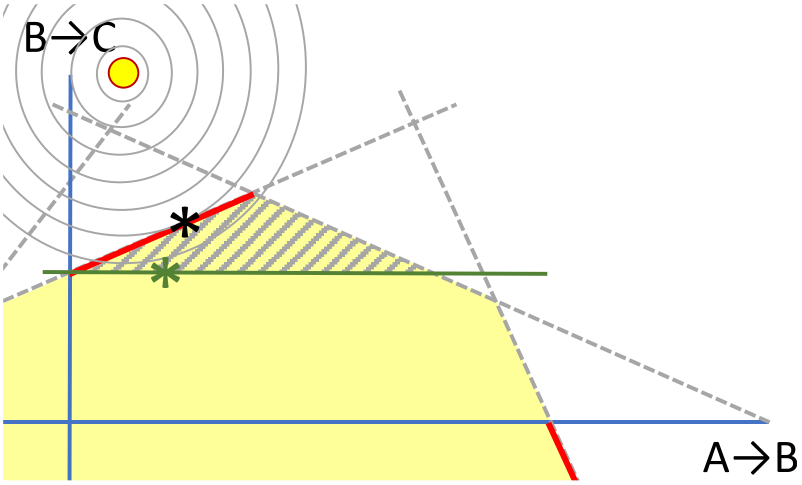
Consider the solution * for case ❶:
Consider the B→C direction: for this direction the price difference must be intuitive: if it were not, the exchanges can be slightly reduced without violating the FB constraints, and without impacting what could be exchanged between A and B.

Consider the A→B direction: although we could decrease the exchange without violating the FB constraints, the decrease would invalidate the B . In fact, if we increase A→B, it would allow us to increase the B→C exchange, which we know has an intuitive price difference, so would contribute positively to welfare. Since we defined * as the optimal solution, it must be so that the A→B price difference is non-intuitive.

Now consider case ❷:

Here for both the A→B and B→C directions decreasing the exchange is possible without violating the FB constraints, nor impacting the exchange on the other direction. I.e. prices will be intuitive for both directions.

With this background, the intuitive constraints that EUPHEMIA generates can be framed in a new context. When EUPHEMIA discards the relieving effects for the directions with the non-intuitive results, we can illustrate this by adding the green constraint, and discarding the shaded region from the valid domain:



The non-intuitive optimal solution (black *) will be replaced by a necessarily intuitive solution (green *). Confirm that for the green constraint solutions must be intuitive.

Heuristic 1

Please note that the introduction of the intuitive constraint discards the shaded region from the solution space. However where we could induce for the red segments of the FB domain solutions must be non-intuitive, the same cannot be done for all the points in the shaded region. It could be that a more optimal intuitive solution than green * exists in the shaded region, but EUPHEMIA cannot identify it. I.e. the intuitive constraint is too strong, and hence shall be considered a heuristic.

Heuristic 2

This paragraph so far focused exclusively on a single hour example and did not consider fill-or-kill orders like block and complex orders. If we consider the branching on fill-or-kill orders EUPHEMIA manages through a branch-and-bound tree, we can imagine that depending where in the tree we are, different intuitive constraints shall apply. Rather than continuously re-evaluating the intuitive constraints that have been added, EUPHEMIA adds these constraints to the global problem. Again this is a heuristic, as the constraints are only locally valid.

Heuristic 3

Finally mind that more than 1 intuitive constraint may be required to find a final intuitive solution. The order in which they are introduced may have an impact on the optimality of the final solution.

Impact of intuitiveness

As the heading of this section suggests, EUPHEMIA supports intuitive FB using heuristics. The impact is that we may lose some optimality of the final solutions.

Moreover, the intuitive mechanism also creates some overhead:

- Whenever EUPHEMIA finds a solution, it shall evaluate if a decomposition into intuitive exchanges exists, or not. This requires a dedicated problem to be solved, which takes some time;
- If no intuitive exchanges exist, EUPHEMIA rejects the solution, which under "plain" FB could simply be accepted;

- Finally the intuitive constraints need to be added to the problem, before resuming the search;

EUPHEMIA supports the FB intuitive requirements. However without this requirement some overhead can be saved, and one can expect some increase in EUPHEMIA performance compared to the situation with the activation of the intuitive requirements.

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 welfare optimization, EUPHEMIA might reject blocks with low volumes that are deep in-the-money, because the impact of these blocks on welfare 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 welfare, under a set of given constraints:

- network constraints
- clearing constraints
- hourly 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[7] to calculate prices and volumes on which welfare calculation is based on.

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 social welfare) is quadratic in terms of the acceptance variables. This is due to the interpolated orders: their marginal contribution to the welfare 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[8].

[7] 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).

[8] 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 social welfare.

4. The coupling problem involves so called primal-dual constraints, i.e. constraints involving both primal and dual variables in their expression[9].
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 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.Welfare 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 welfare:

$$- \sum_{\substack{m,h,s,o: \\ Step\ Orders}} ACCEPT_{m,h,s,o} q_{m,h,s,o}\ p^{O}_{m,h,s,o} \quad (1)$$

$$- \sum_{\substack{m,h,s,o: \\ Interpolated\ Orders}} ACCEPT_{m,h,s,o} q_{m,h,s,o} \left(p^{O}_{m,h,s,o} + ACCEPT_{m,h,s,o}\ \frac{p^{1}_{m,h,s,o} - p^{O}_{m,h,s,o}}{2} \right) (2)$$

$$- \sum_{bo,h} ACCEPT_{bo} q_{bo,h}\, p_{bo} \quad (3)$$

$$- \sum_{m,co,h} ACCEPT_{m,co,h,o}\ q_{m,co,h,o} p_{m,co,h,o} \quad (4)$$

$$- \sum_{mo} ACCEPT_{mo}\ \ q_{mo} p_{mo} \quad (5)$$

$$- \sum_{l,u,h} Tariff_{l,h}\ FLOW_{l,u,h} \quad (6)$$

$$-M \sum_{\substack{m,h,o: \\ Price\ Taking\ Hourly \\ Demand\ orders}} MAX\_CURTAILMENT\_RATIO \quad (7)$$

[9] 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.

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

1. is the contribution of hourly step orders
2. is the contribution of hourly interpolated orders
3. is the contribution of block orders
4. is the contribution of complex orders
5. is the contribution of merit orders
6. is the impact of Tariffs
7. This expression is added to the welfare. 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 welfare. The quadratic function will tend to harmonize the curtailment ratios across the curtailed markets if any

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

Note in case of price indeterminacy where either the lower bound is at minimum price, or the upper bound is at maximum price, a satellite bidding zone, defined as a bidding zone with only simple hourly 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 models that EUPHEMIA solves, to uniquely establish scheduled exchanges at the bidding zone, scheduling area and NEMO trading hub levels respective. See section 6.8.5.

This model shall be compatible with the eventual TSO DA Scheduled Exchanges Calculation Methodology.

Bidding Zone flow calculation

This step aims at uniquely define the flow results between bidding zones, in case indeterminacies remain. It uses linear and quadratic cost coefficients associated to each of the BA lines: EUPHEMIA minimizes the following function:

$$\min\left(\sum_{h}\sum_{l}\left(lc_{l} * \left(f_{l,h,up} + f_{l,h,down}\right) + \ qc_{l} * \left(f_{l,h,up}^{2} + f_{l,h,down}^{2}\right)\right)\right)$$

Where *h* represents the periods, *l* the lines (both ATC and FB), *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:

$$\min\left(\sum_{h}\sum_{sl}\left(lc_{sl} * \left(f_{sl,h,up} + f_{sl,h,down}\right) + \ qc_{sl} * \left(f_{sl,h,up}^{2} + f_{sl,h,down}^{2}\right)\right)\right)$$

Where *h* 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 flow variables to be determined.

Moreover following constraint need to be satisfied to pro-rates cross border exchanges across the underlying scheduling area flows[10] according to installed thermal capacities:

$$\left(f_{sl,h,up} + f_{sl,h,down}\right) = \frac{TC_{sl}}{\sum TC_{sl}} * \left(\overline{f_{l,h,up} + f_{l,h,down}}\right)$$

Where *h* represents the periods, *l* the BZ lines, *ls* the Scheduling Area lines, *up* and *down* the direction of the Scheduling Area or BZ line,

[10] In case both bidding zones only have a single scheduling area, the full bidding zone flow will flow between the scheduling areas.

$\underline{TC_{sl}}$ the Thermal Capacity of the Scheduling Area line and $\overline{f_{l,h,up} + f_{l,h,down}}$ represents flows of the BZ lines being parent of the underlying Scheduling Area lines *sl*.

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:

$$NFE_{A|B} = \sum_{h \in H} \sum_{l \in L_{A,B}} P_B^h * \left(1 - loss_{n_1,n_2}\right) flow_{n_1,n_2}^h - P_A^h * \left(1 - loss_{n_2,n_1}\right) flow_{n_2,n_1}^h$$

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^h, P_B^h is the clearing price for bidding zone of CCP A and B respectively for market time unit h
- $flow_{n_1,n_2}^h$ is the Scheduled Exchange from NEMO trading hub n_1 to NEMO trading hub n_2 for market time unit h
- $loss_{n_1,n_2}$ is the loss associated to the network constraint underlying scheduled exchange, or 0 if no such constraint exists
- h is the market time unit and H is the set of all market time units

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 firstly minimized using a sum of quadratic terms

$$\min \sum_{c \in CCP} \sum_{c' \in CCP \setminus \{c\}} \left(NFE_{c|c'}\right)^2$$

with:
- CCP is the set of all the CCPs
- c is a CCP
- c' is other CCP different than CCP c

4. A second minimization problem is applied using linear and quadratic cost coefficients to avoid any indeterminacies and define a solution consistent with the Scheduled Exchanges between scheduling areas calculated.

$$\min\left(\sum_{i=1}^{n} lc_i * flow_{n_1,n_2}^{h} + \sum_{i=1}^{n} qc_i * \left(flow_{n_1,n_2}^{h}\right)^2\right)$$

with:
• lc_i is linear cost coefficient associated to of NEMO trading hub border i
• qc_i is quadratic cost coefficient associated to of NEMO trading hub border i
• $flow_{n_1,n_2}^{h}$ is the Scheduled Exchange from NEMO trading hub n_1 to NEMO trading hub n_2 for market time unit h
• n is total number of NEMO trading hub borders considered in the optimization, meaning Scheduled Exchange from NEMO trading hub n_1 to NEMO trading hub n_2

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
$NET\_EXPORT\_FB_{z,h}$	net export of bidding zone z,period h flowing on the meshed FB network	Real
$NET\_EXPORT\_LTA_{z,h}$	Net LTA export of bidding zone z at period h flowing on the meshed FB network	Real
α_p:	the level of usage of the FB virgin domain for PTDF matrix p	[0..1]
$FLOW\_FB_{z,z',h}$	Flow on the virtual PTDF line from bidding zone z to bidding zone z' at period h	Non-negative
$FLOW\_LTA_{z,z',h}$	Flow on the virtual PTDF line from bidding zone z to bidding zone z' at period h	Non-negative

Constraints

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

$NET_{POSITION_{z,t}} = \sum_{l:from(l)=z} \left(FLOW_{l,h(t),up} - \left(1 - loss_{l,h(t),down)}\right)FLOW_{l,h(t),down}\right) + \sum_{l:to(l)=z} \left(FLOW_{l,h(t),down} - \left(1 - loss_{l,h(t),up)}\right)FLOW_{l,h(t),up}\right) + NET_{EXPORT_{FB}}{}_{z,h(t)} + NET_{EXPORT_{LTA}}{}_{z,h(t)}$	
$FB\_DEVIATION_{z,h}$: $NET\_EXPORT\_FB_{z,h} = \sum_{z\prime}\left(FLOW\_FB_{z,z',h} - FLOW\_FB_{z',z,h}\right)$	This constraint imposes the balance in the intuitive (virtual or virgin) submodel. It decomposes the net position in terms of the virtual flows.
$LTA\_CAP_{z,z\prime,h}$: $FLOW\_LTA_{z,z',h} \leq \left(1 - \alpha_p\right)LTA_{z,z',h}$	This constraint states that the minimum and maximum LTA in a PTDF line must be respected
$LTA\_DEVIATION_{z,h}$: $\sum_{z\prime}\left(FLOW\_LTA_{z,z',h} - FLOW\_LTA_{z',z,h}\right)$	This constraint imposes the balance in the intuitive LTA submodel. It decomposes the net position in terms of the virtual flows.
$DEVIATION_{z,h}$: $NET\_EXPORT\_FB_{z,h} + NET\_EXPORT\_LTA_{z,h} = \sum_{z\prime}\left(FLOW_{z,z',h} - FLOW_{z',z,h}\right)$ With $FLOW_{z,z',h} = FLOW\_FB_{z,z',h} + FLOW\_LTA_{z',z,h}$	

Annex C.5. Indexes and Annotations

m	Bidding zone
h	Period
s	Supply/Demand
c	Curve identified by *m,h,s*
o	Hourly Order identified by *m,h,s,o*
bo	Block Order
mo	Merit order

po	PUN order
co	Complex Order , where • complex curve is identified by *m,co,h* • complex suborder by *m,co,h,o*
l	(DC/ATC) Line
uu(convention: up=0 and down=1)	Up/Down direction
ACCEPT [0;1]	Acceptance variables
p	Offered Price
q	Offered Volume
MCP	*Market clearing price*