Annex 3.1

Validators Report 1: Evaluation criteria and testing methodology

11-Apr-08

written by:

Alain Billionnet

Ingo Ellersdorfer

Sourour Elloumi

Bernard Fortz

1. The CWE Project and the task of the validators

The CWE project aims at coupling the 4 day-ahead markets of the CWE power exchanges, through a flow-based implicit capacity allocation mechanism. The core of this mechanism will be an algorithm calculating the optimised market results (volumes and prices), from an input consisting of information from the PXs' order books and network parameters provided by the TSOs.

The CWE project has set up a workstream (the "Algorithm Design Workstream"), with the mission of designing the coupling algorithm. The main task of the Validators was to judge on the selection work done by the Algorithm Design Workstream, and to compare the proposed algorithms, both theoretically and based on the results of empirical testing performed by the candidates. This comparison also provides a list of advantages and disadvantages of each algorithm. The validators have to provide together one common report, comprising three parts :

Part I : Evaluation criteria and testing methodology.

Part II : Theoretical analysis of the algorithms.

Part III : Testing results for selection including their final recommendations.

The present report analyses the evaluation criteria proposed by the Algorithm Design Workstream

2. Problem background

The problem under study is the coupling of 4 electricity markets. Usually, the main objective is the maximisation of total net utility. As the capacity for cross-market exchanges is limited, congestion must be taken into account (ATC or flow-based formulation). This might lead to different clearing prices in different markets.

When all orders can be partially filled, the problem is well solved and the solution is unique, with the property that *in-the-money* orders are accepted, and *out-of-money* orders are rejected. This property is a direct consequence of duality theory.

Unfortunately, the markets under consideration also allow block orders that are *"fill-or-kill"*, i.e. they must be completely accepted or completely rejected. This makes the problem a combinatorial optimisation problem, which might be infeasible if both in-the-money block orders must be accepted and out-of-the money must be rejected. To make the problem feasible, in-the-money block orders are allowed to be paradoxically rejected, since paradoxically accepted orders are forbidden by market rules.

With this new problem setting, total net utility might not been seen as the only evaluation criterion: the number of Paradoxically Rejected Blocks (PRBs) should remain small, as well as the amount by which PRBs are in-the-money (*deltaP*).

A solution to the problem is given by:

- the set of accepted and rejected blocks;
- the amount of scheduled electricity exports / imports between markets;
- a clearing price in each market.

Since duality principles do not hold anymore, they are translated into High Level Properties (HLP) that a solution must satisfy (see Document 3).

3. Evaluation criteria

As a preliminary remark, note that two kinds of clearing procedures can be used, independently of the algorithm used for market coupling. The algorithm returns a set of accepted and rejected blocks, the amount of scheduled electricity exports / imports between markets, as well as clearing prices for each market. If clearing on each of these markets is done based on these prices, we talk about *price coupling*. On the other hand, if prices are re-computed in each market following local rules (taking into account blocks and export/import information computed centrally), we talk about *volume coupling*. The choice of the kind of coupling used is independent of the central algorithm, and, in our opinion, algorithms should only be evaluated on the basis of price coupling as volume coupling might be implemented differently in different markets, and the effects on the behaviour of the coupled markets depend on these local policies, not on the central algorithm to be evaluated here.

We are now reviewing the evaluation criteria proposed in Document 4.2.

1. Mandatory requirements

   These requirements must be fulfilled by the selected algorithm. These requirements define

   - the order types that must be handled (hourly orders - stepwise and linear- and block orders);
   - capacity constraints (network flow or ATC-based representation) that must be satisfied by the solution;
   - High Level Properties (HLP): these properties are market rules relating the accepted orders with the market clearing prices; they find their origin in optimality conditions for the problem without block orders.

   One big problem we face is that the HLP as defined now are incompatible. The first rule states that market prices are within pre-defined price boundaries per market, subject to a predefined tolerance. However, it is very easy to construct examples in which this condition is incompatible with the HLP defining congestion prices (see e.g. the example in the document "MLC – Mathematical Model", pages 23-24).

   At the moment, we think that the algorithms SHOULD NOT take price boundaries into account, because it is theoretically impossible to make them compatible with HLP. This does not prevent markets to impose price boundaries on the participants, but the final clearing prices should not be constrained within the boundaries. Another approach would be to redefine the HLP to make them consistent with the price boundaries, but this is definitely out of the scope of the analysis of the algorithms.

2. Optimality and quality of the solution

   2.1. Optimality of the solution: we assume that the objective function considered is Total Net Utility for all algorithms. In the case an algorithm does not return an optimal solution, an interesting indicator, in addition to the gap to the best known solution (as proposed in the document), is the comparison to an upper bound. Such an upper bound is provided by approaches based on branch-and-bound (COSMOS, OMC). For testing purposes, it would be interesting to get an optimal solution in each case even if the 10 minutes time limit is exceeded.

   2.2. Quality of the solution in terms of market aspects: Since Net Utility Value is as important as deltaP and the number of PRBs, the selection should be based on weighting factors. Apart from this, in case several solutions with the same Net Utility Value exist, a solution with a lower deltaP and/or a smaller number of PRBs should be preferred. If otherwise solutions with the same deltaP and/or the same number of PRBs exist, a solution with the higher Net Utility Value should be preferred. An absolute number of PRBs and an upper limit for deltaP that is acceptable within the solution have not been defined. As optimality and quality of the solution are two different requirements to be fulfilled by the algorithms, it is not fully clear how these criteria will be weighted.

   This issue still remains open: we think the weighting of these parameters should be well defined before the testing phase starts, but it seems no agreement is reached yet among the parties. The tuning of the algorithms might depend on this decision, and the testing results might differ significantly following this tuning.

3. Simplicity

   We think this aspect is not relevant for the testing phase of the prototypes, but it will be evaluated within the theoretical evaluation report.

4. Performance

   All candidates should provide evidence that a *feasible* solution can be obtained within the time limit. The numerical precision of the computations performed should be consistent with data precision (e.g. if network flow data are given with 11 digits precision, CPLEX precision should be increased to 11 digits as well to ensure the flow-based constraints are really satisfied).

5. Scalability

   The impact of instance sizes on the quality of the results within the time limit should remain under control. The impact of increasing / decreasing the time limit should also be evaluated.

6. Robustness

   As defined in the algorithm requirements, this concept is only relevant for the final implementation, not during the algorithm selection phase.

7. Reliability

   This will be evaluated within the theoretical evaluation report.

8. Transparency

   The documentation should be complete enough for reproducing the results. Mathematical models used should be well defined. However, due to the complexity of the problem, we think making the algorithm explainable to non-experts might be a very difficult but very important task.

9. Data sensitivity of the parameters setting

   The documentation of the algorithms should clearly state how parameters are set and explain their impact on the quality of the solutions.

10. Extendibility

   This will be evaluated within the theoretical evaluation report, and by scalability tests.

11. Fairness

   Non-discriminatory selection of PRBs. A FIFO rule is currently in use in case of two block orders in-the-money of which one has to be paradoxically rejected, but this aspect depends more on market rules and will not be evaluated during the testing phase.

In Combinatorial Optimization, the behaviour of an algorithm can be very dependent on the size of the problem. It is important to have an idea on the kind and size of "typical" instances that the selected algorithm will have to solve in the future, especially within the limitation of 10 minutes of computing time. The size of an instance could be described by several characteristics:

- the number of markets
- the number of block orders and the number of hours they link
- the number of stepwise simple orders
- the number of piecewise linear simple orders
- the number of network elements

4.Testing methodology

The advantage of the proposed weighting methodology is its simplicity. But it is not clear if it will be sufficient to discriminate the algorithms based on technical, functional and commercial criteria. Regarding the killing criteria, we think that some of the mandatory requirements are not due to the testing methodology (e.g. linear or stepwise hourly orders, "standard" block orders, flow-based representation, ATC) but can be checked within the theoretical evaluation. Other mandatory requirements can only be checked by testing (e.g. calculation time performance, scalability, quality of the solution from the optimality point of view, reproducibility). These killing criteria will be evaluated theoretically (are they potential cases of failures for the algorithms?) and in light of the testing results (do this cases arise in practice or are these only theoretical "pathologic" cases?)

Testing verification will be based on cross-checking of the results obtained on the agreed output data. The checking module of each candidate should be able to work with the input-output data as described in PID DES 4.3. The module should be documented clearly and its methodology should be presented to demonstrate the approach chosen.

For evaluating the scalability it has to be mentioned that the performance of the algorithm not only depends on the computer used, but also on the operating system and the software version (e.g. CPLEX version).

5.Input / output

To evaluate the results and performance of the two different algorithms, it is necessary to have a few more information on various parameters and variables. Additional output needed is:

1. CPU time for performance: This information can be used for comparison of the various algorithms regarding their computational efficiency and solution improvement per CPU time unit
2. Number of PRBs: This information is needed for evaluating the quality (cf. evaluation

criteria 1.2) of the algorithm/solution

3. ID (if possible) of paradoxically rejected blocks: If it is possible to provide information on the specific blocks that are rejected by the algorithm, it could be possible to check if there are blocks or groups of blocks (e.g. depending on the order of the set) that are systematically rejected. Following the last conference call, it seems this aspect is not fully specified yet.
4. Information on deltaP: This information is also needed for evaluating the quality (cf. evaluation criteria 1.2) of the algorithm/solution
5. Optimal/suboptimal solution with bounds (upper and lower bound if possible): This information is needed for evaluating the optimality (cf. evaluation criteria 1.1) of the algorithm/solution.
6. Number of iterations/nodes: Similar to the information on CPU time, the number of iterations/nodes can help to evaluate the computational efficiency of the various algorithms.
7. Evolution of welfare: Tracking the change in welfare of a feasible solution over iterations or over each 1 or 2 minutes during computation can also be helpful to evaluate computational efficiency of the solution found. This information can be linked to CPU time. This information is not available for MLC.
8. Information about price volatility: As price volatility can be of importance for market participants it should be provided for various test settings defined. Update: this is a market aspect, that should not be taken into account in the evaluation of the algorithms.