PowerMatcher Protocol Evolution

Mark Bastiaans (TNO) – January 2013 – v007

The "PowerMatcher Protocol" is a term loosely used to cover a wide range of PowerMatcher-related implementation details. The most apt definition is that it describes the way in which PowerMatcher core components interact. It is specified in the "PowerMatcher - Communication Protocol Specification" document.

During the course of developing the Java version PowerMatcher for Alliander in 2012, two major evolutions were identified:

- **Increasing Bid Accuracy**: originally, bids are limited to a fixed, market-defined number of price steps; the current version of the PowerMatcher supports variable-resolution bids. However, there are still imposed limits on the accuracy of the bid, such as price bounds, price steps and significance.
- **Improving Configuration Protocol**: currently, the Market Basis is shared between PowerMatcher components as a semi-fixed configuration for – amongst other parameters – pricing bounds (minimum and maximum). With a number of changes pending in the configuration parameters needed to run a market, the Market Basis in itself might not the right mechanism for enforcing market-wide parameters.

Both evolutions have implications for the protocol, but also for internal core component behavior. This documents therefore explicitly defines both bid interpretation and representation and recommends a course of action for the PowerMatcher Protocol.

Table of Contents

Bid Interpretation and Representation

Bids are series of power values based on price. They have an important property: they are always *descending* (every consequtive power value is less or equal than the previous value). This is a different approach than the original (C#) PowerMatcher took; it enforced *strictly descending* bids (every consequtive power value is less than the previous value). In practice, this involved a “hack” in the implementation, which is something we wish to avoid in the current and future versions of the PowerMatcher.

Both evolution goals have a shared interest in two important areas that relate to bids: **bid representation** and **bid interpretation**.

Bid Interpretation

The interpretation of bids – the way the bids should be read – is important to the calculation of aggregated bids in Matchers and equilibriums calculation in Auctioneers.

Stepwise and Interpolated

Both **stepwise** and **interpolated** bid interpretation are possible in the current version of the PowerMatcher protocol (Figure 1).

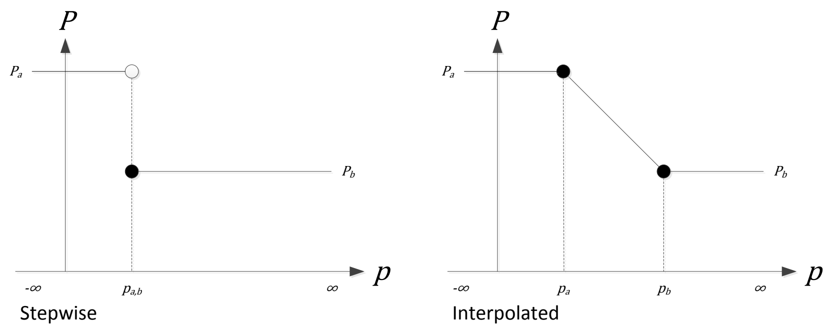

Figure 1 Stepwise and interpolated bid interpretation. Stepwise interpretation uses a mathematical discontinuity to determine power values for prices that have not been explicitly defined in the bid. The leftmost price point in a step is always inclusive; the rightmost exclusive. Interpolated interpretation uses linear interpolation.

In the context of Figure 1, interpretation of a bid $\{(p_a, P_a), (p_b, P_b)\}$ where $p_a = p_b$ implies that:

1. $\forall p\colon -\infty \leq p < p_a : P(p) = P_a$
2. $\forall p\colon p_a \leq p \leq \infty : P(p) = P_b$

When $p_a < p_b$, bid interpretation implies that:

1. $\forall p\colon -\infty \leq p \ \leq p_a : P(p) = P_a$
2. $\forall p\colon p_a < p \ < p_b : P(p) = P_a + (P_b - P_a)\frac{(p-p_a)}{(p_b-p_a)}$
3. $\forall p\colon p_b \leq p \ \leq \infty : P(p) = P_b$

Equilibrium Calculation

The PowerMatcher equilibrium calculation algorithm works by aggregating bid curves and choosing an equilibrium price when the power value on the aggregated bid curve hits zero. This works in most cases; however, there are several special cases to consider when calculating the equilibrium price.

The special cases are partially considered in the current implementation of the equilibrium calculation algorithm; however, this section makes some cases explicit and a revised implementation should reflect these changes.

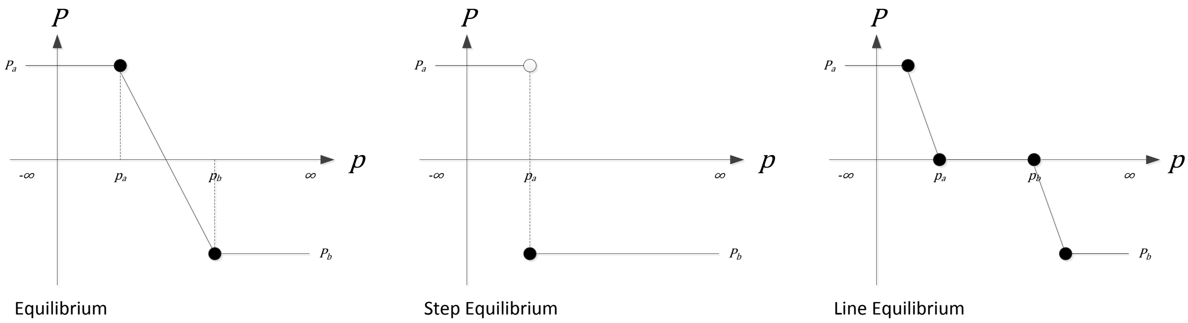

Figure 2 Equilibrium cases. The first case considers a corner case considers a step equilibrium where a power step crosses zero, the second step an equilibrium where multiple equilibrium prices exist and the third case a non-equilibrium, which implies that the bid never reaches zero.

The **equilibrium** case is a true equilibrium: a line segment connected by two points $\{(p_a, P_a), (p_b, P_b)\}$ crosses the zero-power axis at one certain point. Using linear interpolation in the context of Figure 2, this implies the following for the equilibrium price p_e:

- $P_a + (P_b - P_a)\frac{(p_e - p_a)}{(p_b - p_a)} = 0$

The **step equilibrium** case is a specific corner case where a discontinuity crosses the zero-power axis. It is in fact no true equilibrium. In the original PowerMatcher, a price step or a *minimum distance between prices* $\min(\Delta p)$ was used to determine a “fair” equilibrium price. Since in practice issues with significance and rounding make implementation of such a variable impractical, we make a conscious decision not to include this distance and use a fixed equation for the equilibrium price in this case. There are two distinct power values P_a and P_b for which the price can be zero. In the context of Figure 2, this implies the following for the equilibrium price p_e:

- $p_e = p_a$

The **line equilibrium** case is a specific corner case in which if both start and end prices are not infinite the equilibrium price is chosen halfway between the two consecutive price points with power value zero. This implies the following:

1. $p_a \neq -\infty \wedge p_b \neq \infty : p_e = \frac{p_a + p_b}{2}$
2. $p_a = -\infty \wedge p_b \neq \infty : p_e = p_b$
3. $p_a \neq -\infty \wedge p_b = \infty : p_e = p_a$

There are a number of special cases where the aggregated bid curve does not cross the zero-power axis. Additionally, there is a case where there is neither supply or demand. These three cases are visualized in Figure 3.

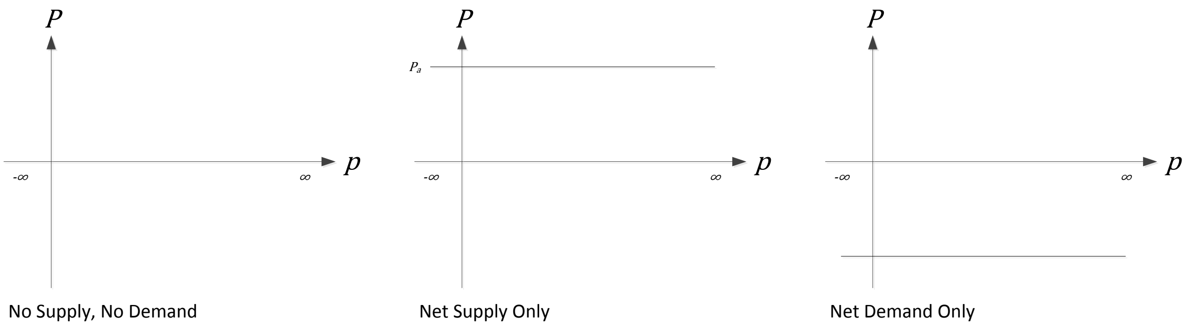

Figure 3 Special supply and demand cases. The first case is a flat-lined bid curve on the zero-power axis. The second and third case involve a net bid curve (either a constant line curve as shown in the figure or a more complex curve) that never crosses the zero-power axis.

The **no supply, no demand** case involves a “flat line” bid curve on the zero-power axis. In this case, we choose the equilibrium price to be zero:

- $p_e = 0$

The **supply-only** case involves a bid curve (either a constant line curve or a more complex curve) which never touches the zero-power axis, and lies above it. In this case, we choose the equilibrium price to be infinity:

- $p_e = \infty$

Similarly, the **demand-only** case involves a bid curve (either a constant line curve or a more complex curve) which never touches the zero-power axis, and lies below it. In this case, we choose the equilibrium price to be negative infinity:

- $p_e = -\infty$

Aggregation and Price Statistics

As stated in the previous section, the Matcher algorithms in the PowerMatcher aggregate the bid of connected Agents. The **aggregated bid** held up-to-date at the level of the Auctioneer Agent can be seen as a merit-order list of all devices participating in the cluster. In order to make optimal coordination decisions involving individual demand-response units, having this list in the right order is of utmost importance. To achieve this device agents need to construct their bid in a certain way:

1. Where the bid can be based on **true marginal (e.g. operational) cost**, a device agent must do so.
2. Where a bid cannot be based on marginal cost, the agent should use the **price history** to determine what ‘low’ and ‘high’ prices are for the market context the agent is in.

Only a few types of agents are able to base their bids solely on marginal cost (point 1) as shown in Figure 4. See [1] for more background info on this topic.

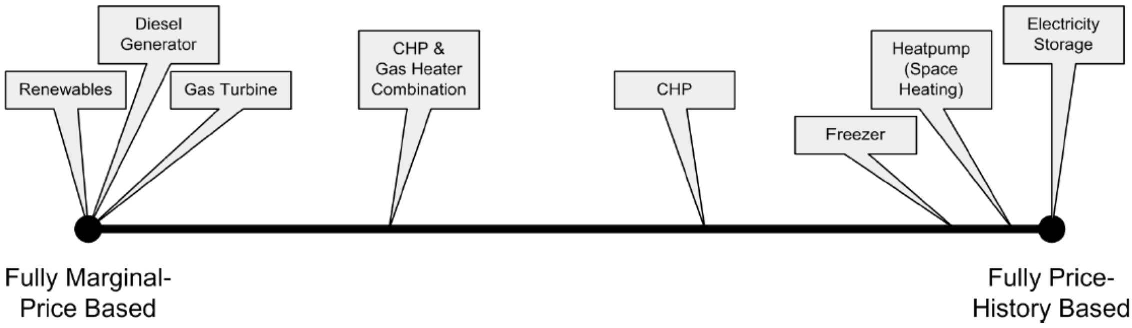

Figure 4 Bid Strategy Spectrum for Distributed Energy Resources based on momentary marginal cost levels [1]

The original PowerMatcher used maximum and minimal price parameters. Through these parameters, the Auctioneer indicated the price range for which it expected a bid from the device agents. However, in many agent implementations these values were treated as the max and min prices to expect. This may lead to sub-optimal operation of a PowerMatcher system. Therefore, an alternative is needed that prevents agents to issue sub-optimal bids.

Apart from the agents that are able to base their bids on marginal prices alone (the ones in the left side hand end of the spectrum in Figure 4), agents need to develop a notion of what is cheap and expensive in their market environment and given their operational characteristics. To derive appropriate levels of ‘high’ and ‘low’ prices an agent must either:

1. build-up **own statistics** on the price history, or
2. receive statistical parameters from an **outside service**.

Bid Representation

The representation of bids – the way data structures resemble the bids – is important to the interaction between PowerMatcher components. Representation has repercussions for the ability to communicate: for example, an intricate representation may incur heavier costs on bandwidth than a simple one.

Bid Data Structure

The original PowerMatcher represented bids with a simple indexed array, which implies a **fixed resolution** (Figure 5). This means that every power value in the series is linked to a price which is always a fixed distance away from the previous and / or next price. The resolution is influenced by the *price step* and the amount of prices allowed. Both values are set in the Market Basis and bid data structures.

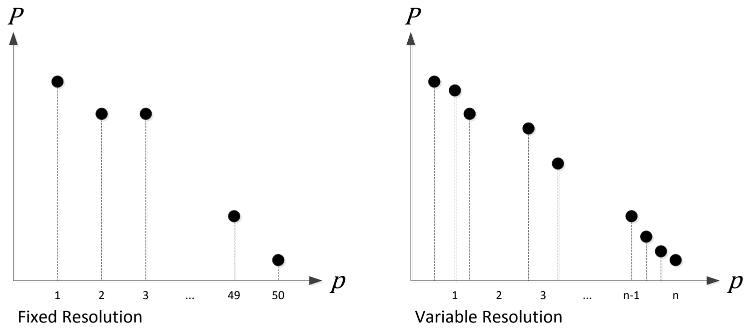

Figure 5 Fixed-resolution and variable-resolution bid representation. Fixed-resolution bids have a fixed number of price points and price points are set at fixed price steps; variable-resolution bids have potentially unlimited price points and variable price steps.

An alternative approach is representing the bids with an collection of points $[(p, P), \ldots]$, which implies a **variable resolution** (Figure 5). This implies letting go of fixed price steps and fixed number of price points. This approach is also known as the "line segment" approach. Both fixed and variable-resolution bids are supported in the current version of the PowerMatcher Protocol.

Both approaches influence data structures and the complexity of algorithms (both bid aggregation and equilibrium calculation) in the implementation. While complexity is often approached theoretically with sound practical big-order mathematics, the practical implications of both approaches must be carefully considered (Table 1).

Both the HeatMatcher project and the NXP PowerMatcher protocol for the HAN implement variable-resolution bids. A variable-resolution protocol has already been implemented in the current version of PowerMatcher.

Table 1 Bid and algorithm complexity for fixed and variable-resolution bids

Resolution	Bid Data Structure	Worst-case Algorithmic Complexity	Practical Implications
Fixed (original)	Indexed array of P with fixed size n	$\boldsymbol{O}(n \times m)$, with n the bid size and m the amount of bids	None (original situation)

Resolution	Bid Data Structure	Worst-case Algorithmic Complexity	Practical Implications
Variable (current)	Collection of (p, P) with arbitrary size	$\boldsymbol{O}(L)$ with L the total amount of elements in all bids	• Bid curves are usually not very intricate and can be represented with less elements than in a fixed resolution. • The algorithms can be optimized based on the descending properties of bids and / or bid interpolation. • Encoding schemes (conversion specifications for data types to bits and vice-versa) might limit the maximum length of the array; however, this constraint is not part of the protocol.

Bid Data Types: Significance and Infinity

Significance

A data structure is a mathematical construct; in practice, bids have to fit within certain data types. Data types are bounded by implementation and introduce certain constraints that may affect bid interpretation. One of these constraints is **significance**: the amount of significant digits in which a number (such as a price or power value) is represented.

As stated, significance is important to both interpretation and representation. Data types are always bounded by implementation. For instance, a 64-bit double-precision floating point number can contain a maximum of 16 bits of significance. This has consequences for both the encoding scheme (conversion specifications for data types to bits and vice-versa) for price and bid data sent over a network connection and for the storage of (intermediate) results for the aggregation and equilibrium calculation algorithms.

The significance constraint introduces a number of requirements to both representation and interpretation:

- A general rule of thumb is that the implementation of the algorithms supports *the same order of significance* than is available in encoding scheme(s) which the Matcher has to deal with.
- *Rounding* will be critical to the fairness of the Matcher algorithms. An established method of rounding which is relatively unbiased towards all parties involved (suppliers and demanders in the PowerMatcher case) is banker's rounding, which is the default rounding method used in IEEE 754.

Infinity

Aside from significance, another constraint is placed on the data types. The bid interpretation notes that (minus) **infinity** prices and power values should be possible; the data types and encoding schemes for prices and bids should reflect this. There is a "but": this special value should not be used in calculation, since this will produce unworkable prices.

Recommendations for PowerMatcher Protocol

The following section contains recommendations for improvements to the current PowerMatcher protocol. It is split into recommendations for bid interpretation, bid and price representation and Matcher configuration.

Bid Interpretation

Specify Algorithms

The first section of this document contains example specifications of how the algorithms for bid aggregation and equilibrium calculation work. Such a specification should be incorporated in the current PowerMatcher protocol: all components implementing this protocol will have a general idea of how the core Matcher algorithms work and have a fair chance in a typical market.

Specify Significance

The first section of this document discusses the subject of significance. A very clear set of rules regarding significance should be laid out in the protocol. This can be achieved by clearly specifying how components should handle bid significance internally (i.e. for calculations). This includes specifying standards for associated subjects such rounding. We recommend that banker's rounding should be used in bid aggregation and equilibrium calculation for the sake of fairness.

Bid and Price Representation

Distinguish Between Protocol and Encoding Scheme

The current PowerMatcher protocol specifies two protocols: one for broadband and one for narrowband applications. The fundamentals underlying the protocols are the same and the split between protocols was driven by constraints imposed by application-specific factors such as network connections and power consumption. It is therefore recommended that a distinction is made between **protocol** (the fundamentals behind component interaction and behavior) and application-specific **encoding schemes**. Such schemes should contain information such as:

- The conversion of messages into bits, and vice versa.
- Data type specifications related to the conversion, including the effects of these specifications on factors such as significance.
- Methodology for incorporating a variable resolution (e.g. for price points in a bid).

The current two protocols (broadband and narrowband) can be easily mapped to such encoding schemes.

Incorporate Significance and Infinity

The current protocol has to be changed in two important ways:

- As noted before, **significance** must be taken into account when choosing the data types and encoding schemes for price and power values. The maximum significance in use in encoding schemes for prices and / or power values is the maximum significance usable in bid aggregation and calculation.
- **Infinity** should be incorporated in both protocol and encoding schemes; both positive and negative. This can explicitly be defined as a special character or as part of a custom data

type. It is part of the standard IEEE floating point specification; encoding schemes with different number specifications should take these special values into account.

Matcher Configuration

The **Market Basis** serves as a configuration container for market parameters. A PowerMatcher **market** is defined as an Matcher and all its downstream components (Matchers and Agents). As stated, there is a need to remove bounds-specific parameters from the market basis, and there are voices to remove the Market Basis altogether as a market-wide configuration mechanism.

At the time of writing, the Market Basis contains a number of configuration parameters for a PowerMatcher market (Table 2).

Table 2 Market Basis parameters

Parameter	Definition	Representation	Example	Remarks
Commodity	The commodity being traded on the market	String	`Electricity`	Applies to the whole market
Currency	The currency to trade in	String; ISO 4217	`EUR`	Applies to a Matcher and all first-level Agents; Matcher is responsible for currency conversion
Price Step	Minimum distance between price values in bids	Integer	`50`	Influences the granularity of rounding in equilibrium calculation and aggregation
Minimum Price	Minimum price in bids	Double-precision Floating Point	`-111.11`	Bounds the price to a specified minimum
Maximum Price	Maximum price in bids	Double-precision Floating Point	`111.11`	Bounds the price to a specified maximum
Significance	Significance of prices and bids sent over the market in number of digits	Integer	`3`	Influences the granularity of rounding in equilibrium calculation and aggregation
Market Reference	Market basis identifier	Integer	1	Used as a context for prices and bids, ie. under which market basis (version) a price or a bid should be interpreted

Keep On Piggybacking

Currently, the MarketBasis is sent to Agents as part of a price update message. There are two possible methods for sending configuration data, both with their respective practical implications (Table 3). Since the current method seems to work, no protocol change is recommended.

Table 3 Configuration methods

Method	Description	Practical Implications

Configuration Protocol	Separate protocol for market configuration	• The PowerMatcher protocol becomes more complex • Price update messages are slightly smaller in size • Agents cannot bid before receiving a configuration message
Piggybacking	Configuration parameters from Matchers "hitch a ride" on price update message	• The PowerMatcher protocol is simplified • Price update messages are slightly larger in size • Agents cannot bid before receiving a price update message

Introduce a Price Duration Curve and Price History Service

The previous section listed the necessity for certain Agents to be able to use price statistics, either via an own mechanism or an outside service. The former can be built up by the device agent itself. The latter may be a sophisticated statistical analysis engine providing in-depth and comprehensive information tailored to the specific PowerMatcher application (e.g. virtual power plant operation on the balancing market).

As the requirements to such a communication depends heavily on the specifics of the application, there is no necessity to include it in the standard protocol. However, there is a class of simple agents that run on low-processing-power embedded processors behind a low-bandwidth communication channel. In these cases, there is neither enough processing power and memory to build-up own statistics, nor enough bandwidth to receive comprehensive statistical data. For this class, a simple standard set of statistical parameters that is obtainable from the connected matcher will be desirable. This set should provide means to derive strategic price levels (e.g. for 'cheap' and 'expensive') for a range of devices. As the characteristics of specific devices has an influence on these price levels, the info must be general enough for the device agent to derive its own.

Candidate for such a data structure is a **price duration curve**, i.e. a sorted histogram of all prices over a given time period. The curve consists of a series of 2-tuples (Price, Utilization Rate), where the Utilization Rate is a 0-100% value. The price duration curve should replace minimum and maximum price parameters.

Aside from the price duration curve, Agents will need a notion of price history after initiation. Since some agents will start with a blank slate, a Matcher should include a simple **price history service** that gives an extended price duration curve.

Send Localized Commodity, Currency and Price Duration Curve

As seen in Table 2, not all parameters are relevant. Additionally, parameters are only relevant per layer in the PowerMatcher tree and are thus localized and Matcher-specific. A typical Matcher should only transfer the following parameters:

- **Commodity and Currency.** These values are immutable and only have to be transferred once. Commodity holds for the entire market; currency applies to a Matcher and connected first-level Agents. Matchers are responsible for any currency conversion when using a different currency than the top-level Auctioneer. These parameters are currently not used but should be included for future use and should remain there pending discussion.
- **Price Duration Curve.** See the previous subsection for more details.

The following parameters are not needed:

- **Price Step.** Since we deprecated all uses of the original price step, this parameter can be removed from the Market Basis.
- **Minimum and Maximum Price.** These parameters have been replaced by the price duration curve.
- **Significance.** Since we have specified rounding conventions and price step is currently not necessary in the step equilibrium corner case, significance does not have to be explicitly communicated. It is however explicitly part of an encoding scheme.
- **Market Reference.** This parameter is only used for versioning purposes. Since the only parameter that changes regularly is currency, we recommend that currency is included in every bid made.

References

[1] Koen Kok. Short-term economics of virtual power plants. In *Proc. 20th International Conference on Electricity Distribution CIRED*. IET-CIRED, 2009.