To (Spread) Capture Or Not To (Spread) Capture

The post-trade report has become a key tool for modern trading desks. It provides the desk with a variety of statistics about the performance of the trading tools it employs and the decisions it makes. For example, reports typically include the average arrival price shortfall, the average interval VWAP shortfall, and more. Over recent years, the spread capture metric has become popular, particularly for evaluating algorithmic trading providers. Spread capture purportedly shows the average price improvement the algorithm was able to achieve by using limit orders to provide liquidity rather than crossing the spread to take. Spread capture is usually quoted in percentage of the bid-ask spread, where 0 percent means the algorithm always paid the offer (when buying) and 100 percent means the algorithm always executed at the bid. The popularity of this metric reflects a widespread assumption that the higher the number the better the performance of the algorithm. In this short note we highlight a fundamental problem with the spread capture metric, and show how this metric can give the impression that an algorithm performed superbly when in reality it performed poorly.

Perhaps the best way to illustrate the problem with spread capture as a performance metric is through an example. Consider a VWAP buy order that requires a hundred shares be purchased at 10:00 am, at which time the NBBO is 10.45x10.46. A typical trading algorithm might post a limit order to buy the hundred shares on the bid (10.45). Now suppose that after a few seconds the NBBO changes to 10.46x10.47, and the algorithm re-pegs the limit order to the new bid at 10.46. After a few more seconds, the NBBO changes to 10.47x10.48 and again as a response the algorithm re-pegs its limit order to the new bid at 10.47. Finally, the limit order gets executed at 10.47. What is the contribution to the spread capture metric of this child order? The common approach is to count the spread capture for this execution as 100% because the execution occurred at the bid. However, from a true execution quality standpoint, in hindsight the trader would have been better off crossing the spread and buying at the offer of 10.46 that prevailed at 10:00, though such an execution would contribute 0% to the spread capture metric. Figure 1 illustrates this example.

This example highlights the fundamental problem with the spread capture metric. An algorithm that has a higher spread capture might be inferior to a clever algorithm that provides less liquidity but achieves better shortfalls.

An improvement would be to measure the shortfall relative to the offer price at the time of original order submission, not at the time of execution. In our previous example, the improved metric would be -100% as our execution price was 1c

HIGHLIGHTS

- THE PROBLEM WITH SPREAD CAPTURE METRIC
- IMPROVED "EFFECTIVE" SPREAD CAPTURE
- ORDER SUBMISSION AND PEGGING STRATEGIES

An algorithm that has a higher spread capture might be inferior to a clever algorithm that provides less liquidity but achieves better shortfalls.

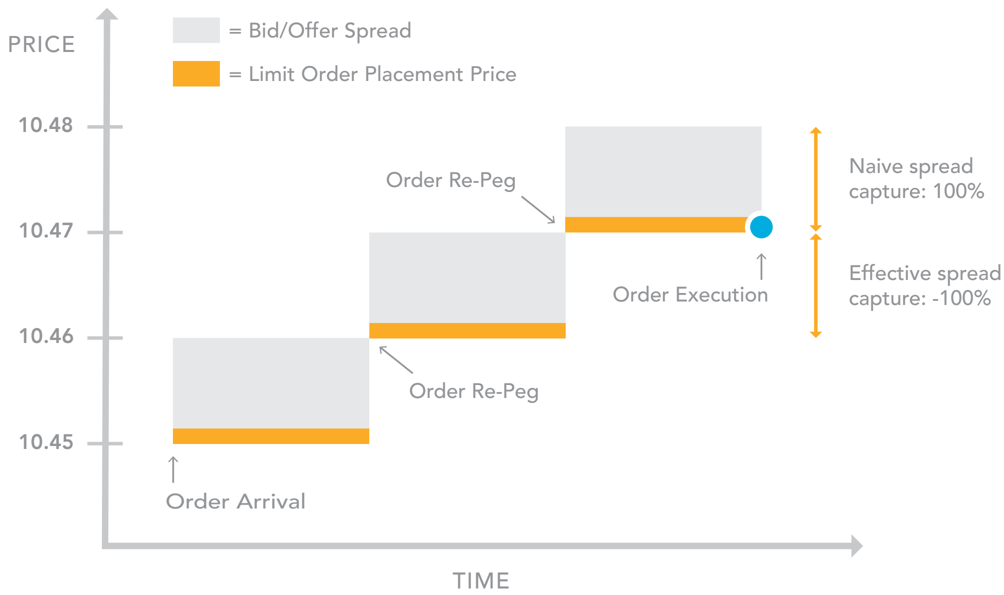

FIGURE 1 An order life cycle example. The order executes well above the offer at the time of submission. The naïve spread capture is 100% while the effective spread capture is -100%.

(one bid-ask spread) higher than the ask price at the time of submission. Measuring spread capture relative to the time of original order submission provides more information on the effectiveness of the execution strategy than measuring the spread capture relative to the time of execution.

However, even this improvement leaves nuances that might make the metric deceptive. When the need to re-peg an order arises, an execution algorithm has two options. The first is to send a cancel/replace message to change the limit price of the order. The second is to cancel the old order and send a new one with the updated limit price. While the two methods have essentially the same end result, the second option can lead to an over optimistic spread capture metric, since the link between the original order and the eventual execution is severed. Similarly an algorithm can at any time "change its mind" and simply cancel a limit order, for example in order to cross the spread, again potentially erasing any loss that occurred during the life of that order from the spread capture metric. These nuances are not purely of academic concern. In our work doing transaction cost analysis we have observed or deduced significant variability in the ratio of new orders to executions among different providers. The effect is that some providers, by sending new orders, could unintentionally obscure the spread capture that their algorithms really achieve.

We have developed a metric we call "effective spread capture" to get at the intention of the spread capture metric in a way that is not subject to the problems described above. Effective spread capture measures the price of each execution relative to the offer price at the time of the first order placed that corresponds to that execution. That is, every execution is matched with the earliest point in time when that order quantity was posted, and the shortfall between the ultimate execution price and the offer at that time expressed in terms of the spread. In our example, regardless of whether the algorithm sends a modified order or cancels and then replaces the order, or simply cancels the order to lift the offer at 10.47, the results should be the same, namely -100%. Effective spread capture is a far more reliable, useful characterization of algorithmic execution quality, and can be useful comparing the performance of providers regardless of order submission and pegging strategies. Unfortunately, it can only be implemented by an algorithmic vendor, or by providing a client or third party with a full child order audit trail, including the placement of unexecuted orders.

Nevertheless, the buyside would be well served by wide-spread reporting of this metric. While reported spread captures frequently top 70%, one can see from first principles that under efficient market assumptions, a trader who is required to complete his order should expect an effective spread capture of zero. In practice, we believe negative effective spread capture algorithms are not uncommon, perhaps because of undue focus on rebates rather than execution quality. A positive spread capture indicates a high quality algorithm that adds value or alpha to the execution.

For questions or comments please email Dr. Eran Fishler, Director of Research (technotes@pragmatrading.com).
Pragma provides comprehensive broker-independent trade cost and trade process analysis services.