Efficient monitoring of financial orders with agent-based technologies

Philippe MATHIEU and Olivier BRANDOUY

Abstract The execution of orders on stock exchanges is managed by a set of formalized rules based on price and time priority. Nevertheless, orders issued by investors do not show-up directly in the market system : they transit through the brokerage intermediation where they can be arranged in different sequences. We show that the latter operation has a critical impact on investors. In this paper, we propose a decision support system that solves the underlying optimization problem for a given social welfare. We show that the solution cannot be obtained without an agent-based simulation platform that individualizes the consequences of the broker decision in terms of order sequencing at the agent (client) level. In this framework, we study the impact of several social welfares functions and show how the broker can grant his clients with *"just and equitable principles of trade"*.

Introduction

In modern stock exchanges, investors almost never have the ability to route their orders directly to the market; they must use the services of a financial intermediary whose task is to trade securities on behalf of his customers: a broker. Thus, the role of the broker consists in introducing their orders in one of the various trading platforms that are available for the relevant financial commodity. These platforms usually run with electronic order books. Basically, an order book captures the on-going continuous double auction process

Philippe MATHIEU
Université Lille 1, Computer Science Dept. & LIFL (UMR CNRS-USTL 8022), e-mail: `philippe.mathieu@lifl.fr`

Olivier BRANDOUY
Sorbonne Graduate School of Business, Dept. of Finance & GREGOR (EA MESR-U.Paris1 2474), e-mail: `olivier.brandouy@univ-paris1.fr`

allowing negotiation between buyers and sellers. Electronic order books implement a "price" then "time" priority system. Sell (buy) orders are organized by increasing (decreasing) prices and, in case of equal prices, placed in the queue with respect to the time-stamp indicating when they were introduced in the book.

Consider the situation of a broker having a set of orders $O : \{o_1, o_2, ..., o_n\}$, awaiting to be inserted in a given order book. Whatever the time at which these orders were issued or whoever the customer are, these orders are pending at the broker's desk, and just about to be introduced in the market and processed by the matching algorithm.Nevertheless, nothing is clearly established concerning the "how-to" introducing orders in the book, so to behave smartly and fairly with respect to clients own interests. Nevertheless, a body of literature, both from Finance and Computer Science has tackled related questions such as orders cost of execution (see for example, [8], [2])[1].

This paper investigates this question, shows why it is of main interest both for Computer Scientists and Financial professionals and how it can be solved in an agent-based Decision Support System. We demonstrate that a Social Welfare Function must be defined by the broker and guaranteed throughout the trading process to achieve just and equitable principles of trade he owes to his customers. We also show that a simple "first-in first-out" principle to rule the order flow does not necessarily lead to an optimal situation for broker's clients. To illustrate the research question tackled in this article, let's consider a simplified situation where an order book has two limit orders listed in the Bid and the Ask queues (see Table 1).

	RAP*	Price	Qty
Ask	2	120	5
Ask	1	110	10
Bid	1	90	10
Bid	2	85	5

* RAP : *rank in the auction process*

Table 1 Initial Order Book

We now show that when a set of orders must be introduced in this order book, the sequence along which they are submitted affects the resulting price sequence. For that purpose, we consider the following limit orders:[2]

- Agent "A" wants to sell 25 stocks at a minimum price of 85.
- Agent "B" wants to buy 10 stocks at a maximum price of 95.

[1] Most of the time, this question is *officially* solved by a "first-in first-out" principle : orders are treated on the fly. Nevertheless, orders can be re-arranged, and sometimes mixed with the brokers own orders (proprietary trading) in "front running" illegal or parasitic operations [4].

[2] A "limit order" fixes the max (min) acceptable price at which an investor agrees to buy (sell) a financial commodity.

The initial situation of these agents is summarized in Table 2.[3]

If the orders are introduced by the broker in opposite sequence (A *then* B) or (B *then* A), the impact on both agents wealth is clearly different due to priority rules and to the fact that the final portfolio is evaluated *mark-to-market* as well (*i.e* "with the last settled price on the market").

		Initial				Final				
	Agent	Cash	Stocks	Price	Wealth	Cash	Stocks	Price	Wealth	Δ
(A *then* B)	Ask	375	25	105	3000	2550	0	85	2550	-450
	Bid	3000	0	105	3000	2150	10	85	3000	0
(B *then* A)	Ask	375	25	105	3000	2650	0	85	2650	-350
	Bid	3000	0	105	3000	2050	10	85	2900	-100

Table 2 Situations for Traders A and B

Therefore, it is obvious that the sequence chosen by a broker when introducing a series of orders is critical for his clients' wealth. Note again that if a time-stamp constrains the Broker (for example order A arriving first to the desk), this does not really solve the problem since it automatically fixes the order execution (first-in, first-out) without considering any aggregate utility criterion. This could be even more discussed when the time-stamps are close in time. If one desires to respect just and equitable principles of trade this implies to avoid decisions that favor one customer (*in casu* A or B) against the others. In this example, the broker faces a *set of combinations* $\aleph = \{0_1, 0_2\}$ with two *vectors of orders* $0_1 = \{A \text{ then } B\}$ and $0_2 = \{B \text{ then } A\}$ Note that the number of combinations of orders (each one generating a specific sequences 0_j) is fairly large, as soon as n becomes important: $n!$. Thus, the broker has to choose in $\aleph$ a Pareto-optimal vector $O{*}_j$ for his customers. This issue is clearly related to a social choice problem [1] and the solution necessitates to define a particular Social Welfare Function (here-after SWF).

This paper is organized as follows. In section 1, we introduce the social welfare functions used by the agent-based decision support system. In section 2, we briefly present the decision support system itself. In a last section (3) we propose a limited set of experiment run over artificial data to illustrate the effectiveness of the architecture.

1 Individual and Social Welfare

Individual actions : One should distinguish two different individual actions with respect to the category of agents: those made by investors and those

[3] We have fixed an initial arbitrary price for the stocks of 105.

made by the broker. Agent's individual rationality (whatever their role is) is supposed to be standard, *e.g.*, they always prefer having more than less.

1. Investors actions:
   - Investors are represented by a population of n autonomous agents. Each of these agents send one *limit orders* o_i to the broker.
   - A limit order is a 4-uplet o_i : {id, direction, price, quantity} determining the name of the agent, if he wants to sell or buy, at which limit price, and for which amount.
2. Broker actions:
   - The broker must route the resulting set of orders $O : \{o_1, o_2, ..., o_n\}$ to the order book. This necessitates deciding in which sequence these orders will be arranged prior to be introduced in the market system. Any combination of all the elements in O is a vector denoted O_j. As mentioned previously, there are $n! = m$ such combinations.
   - We denote $\aleph : \{O_{j,j\in[1,m]} : O_1, O_2, ..., O_m\}$, the set of all possible combinations of orders out of O.

Ideally, investors should obtain as much as they can from their trading activity. Nevertheless, sellers and buyers have clearly opposite interests. Concerning the broker, one should posit that his own aim consists in maximizing the fees he charges his customers with. These fees are based on the market capitalization exchanged by these latter through his services (*i.e.* price $\times$ quantity).

1. Investors welfare (w_i):
   - Let $w_{i,t}$ be the wealth of agent i at date t. For the sake of simplicity, we posit a uniform transformation of agent's wealth in terms of cardinal utility (any kind of Von Neumann utility function [7] can be chosen here like $u(w) = log(w)$ or $u(w) = w^{\alpha}$).
   - $w_{i,t} = p_t.A_i + c_{i,t}$. In this equation, p_t is the price vector for the assets A_i held by agent i (these assets being represented by their quantities); $c_{i,t}$ is agent i cash at time t. Among two states of the world Ω_1 and Ω_2, if $w_{i,t}|\Omega_1 > w_{i,t}|\Omega_2$ then $u_{i,t}|\Omega_1 \succ u_{i,t}|\Omega_2$.
   - $\Delta w_{i,t} = w_{i,t} - w_{i,t-1}$ is the variation of agents individual welfares.
2. Broker welfare (w_B):
   - Let r be the transaction cost applied by the broker and charged to his clients.
   - Any transaction leads to a transfer of cash ($p \times q$) from the buyer to the seller. Therefore, the transaction costs apply on these values cumulated over time. We posit that r is the same for any of the brokers' customers and that fees are charged once for the seller and once for the buyer (see algorithm 1).

```
for (i in 1:n) do
    process $o_i \in O_i$;
    $w_B \leftarrow w_B + 2 \times r \times (p \times q)$
end
```

Algorithm 1: Calculation of Broker's welfare

Social Welfare : We have shown that depending upon the sequence O_i for processing the orders, agents' utilities will be affected accordingly. Based on these individual utilities, a social welfare function (here after SWF) can be defined and used to compute a social welfare measure for each O_i. Let: $U = f(u_{i,i \in [1,n]})$, the collective utility function based on agents individual utilities. This function should ideally respect a series of properties: Extensive developments around these concepts can be found in [6].

Thus, a crucial question here is to determine a computable function U to decide which sequence $0^*_j \in \aleph$ should be chosen in order to ensure a fair treatment of orders among customers.

```
agentsUtilities← NA ;
BestSW← NA ;
BestSeq← NA ;
for (j in 1:m) do
    process $O_j$ ;
    agentsUtilities ← $u^j_{i,i \in [1,n]}$ ;
    sw ← $U(agentsUtilities)$ ;
    if (sw ¿ BestSW) then
        bestSW=sw; BestSeq=$O_j$
    end
end
$O^*_j$ ← BestSeq ;
return $O^*_j$
```

Algorithm 2: Algorithm for the choice of $O^*_j \in \aleph$

One can remark from algorithm 2 that U is run over a matrix of m vectors of n values. There is obviously no unique possible function or process U that can solve the question of the fair treatment of orders and each alternative should be considered carefully. For example, the broker can choose several traditional options. These options are based on real-valued functions delivering a "welfare score".

1. Utilitarian SWF.
   In this case, the broker would choose the vector of orders delivering the highest sum of individual utilities : $0^*_j \equiv max_j(\sum_{i:1}^{n} u^j_{i,i \in [1,n]})$ In the initial example presented in section , this criterion delivers a score of 5550 whatever the sequence. Hence, both are equal and this criterion does not help to choose among these results.

2. Egalitarian SWF (Rawl's "difference principle", see [10]).
   The choice of this rule implies to opt for the sequence of orders delivering the maximum of the minimum individual utility amongst the agents $0^*_j \equiv max_j(min_i(u^j_{i,i\in[1,n]}))$. In the basic example, if this SWF is chosen, one should decide to process B *then* A (max of min values = 2650).
3. Elitist SWF.
   The sequence of orders delivering the maximum individual utility amongst the agents will be chosen : $0^*_j \equiv max_j(max_i(u^j_{i,i\in[1,n]}))$. This approach delivers an opposite outcome (A *then* B, max of max values = 3000) with respect to the egalitarian approach.
4. Nash SWF.
   This approach proposes to consider the sequence of orders from which the product of individual utilities is maximum : $0^*_j \equiv max_j(\prod^n_{i:1} u^j_{i,i\in[1,n]})$ In this last case, B *then* A must be chosen (product of values = 7.685E6).

In our context, these alternative measures can provide a justification for any of the possible combinations in the order sequence. Nevertheless, one can imagine that client A or B, will not appreciate these justifications on the same footage, some appearing fair and others unfair with regard to each individual point of view. Furthermore, notice that as soon as one uses the Egalitarian or the Elitist criterion, one client is favored to the detriment of the other. Nevertheless, this choice could be made by a broker for marketing reasons or to maximize his own utility w_B. The Utilitarian criterion does not help to choose in this particular case, even if it is intuitively appealing in terms of fairness. In summary, in our example, whatever the final decision, A or B will be favored. Note that in the preliminary example, for the sake of simplicity, we have arbitrarily fixed initial identical endowments for the agents. In most of the cases, this will not be true. Thus, the result of the sequence in which orders are entered in the system must be appreciated "as if" agents were wealth-less (without any stock or cash *prior* to trading). This *"wealth-less instantiation"* is an initial, necessary step in the process. It means that an agent willing to sell will have, by construction, negative stock holdings and positive cash after trading, while an agent willing to buy will have positive stock holdings and negative cash. Thus, the Utilitarian and Nash Social Welfare Functions are not adapted because one can imagine situations in which the post-trading wealth of an agent is negative. [4] This calculation, based on a *"wealth-less instantiation"*, has two straightforward consequences: i) the Utilitarian criterion is constant whatever the chosen permutation of orders, ii) the Nash criterion cannot be used due to the possibility of negative individual welfares in some cases. One should therefore consider alternative SWF.

[4] For an application of these criterion in a MAS, see [9].

2 Decision Support System

In this research, we provide a system that allows a broker to simulate within a given order book the execution of his customers' orders in different sequences. Consequently, this decision support system must be able to capture the current state of a real-world market at time t, and to test in a virtual environment where the real-world rules are cloned, the impact of any sequence $O_i \in \aleph$ with regard to the social welfare of the customers population. In doing so, the decision support system extrapolates the impact of the broker possible decision and allow him to select the best possible outcome. Notice again that a social-welfare rule has to be defined prior to any simulation. Thus an agent-based artificial stock market, able to simulate these outcomes, appears vital in this design. Ideally, it not only should allow defining behaviors at the individual level, but also should permit to track the consequences of agents actions at the micro-level, with some realism. To build a powerful decision support system, a full agent-based ASM is definitely necessary since the calculation of social welfare is done from agents' individual welfares. Therefore one should use a system that individualizes the consequences of each agent's action, implements an asynchronous order book and reifies the agents.

The details of the agent-based artificial stock market (here after ABASM) used in this research, named ATOM, can be found in [5]. This platform is validated by both the industrial and scientific worlds: for example, the platform can simulate, using artificial behaviors, the main stylized facts usually considered as necessary for realistic simulations of financial motions (see for example [3]); the "replay-engine" of this platform can use real-world orders and will deliver, in this configuration, the same results as the NYSE-Euronext system . Using this ABASM, a broker will be able to decide, with regard to a social welfare optimization rule, which sequence should be run by the broker in the real-world book. Note again that the choice of a given sequence of orders among all possible sequences cannot be done without tackling its impact at the individual and the collective levels. This imposes to use, in addition to the ABASM itself, a ranking algorithm for each possible sequence of orders.

Ranking Algorithm : The only solution we have found so far is to test all possible combinations of orders. Nevertheless, an initial simplification can be done.

Let b_i and a_i be the sub-vectors respectively gathering the "buy" and "sell" orders ($b_i \bigcup a_i = O$). Let b'_i and a'_i be the ("buy" and "sell") orders yet present the order book. The subset of orders that are critical to determine the optimal social welfare is :

$$O' = \{\{b_i \geq (min(a_i) \vee min(a'_i))\} \bigcup \{a_i \geq (max(b_i) \vee max(b'_i))\}\} \qquad (1)$$

In equation 1, the operator $\geq$ applies on prices appearing in orders. For the next developments, O is supposed to be reduced to O'. In the ranking algorithm 3 presented below, "ATOM" refers to our agent-based artificial stock market presented previously.

```
Input: Order Book (Real Orders), Waiting Orders, R (rule to optimize)
$U$(optimal Social Welfare) $\leftarrow -\infty$;
$O_i^*$ (Optimal Sequence of Orders) $\leftarrow \emptyset$ ;
forall the possible Permut SEQ of Waiting Orders do
    init ATOM with Order Book;
    execute $SEQ$ in ATOM;
    compute SW according to R;
    if $SW > OSW$ then
        $OSQ \leftarrow SEQ$ ;
Display $OSQ$
```

Algorithm 3: Optimal Sequence of Orders

3 Experiments and discussions

In this section, we illustrate the effectiveness of the agent-based decision support system using artificial sets of orders that will be used in a context where the broker has to run the sequence of orders in an empty order-book. This means that each of the broker's orders is matched internally (within the set of broker's clients orders). This is a limit case known has *"systematic internalization"*. We have chosen to generate artificial data for the sake of simplicity and tractability of the results. The following algorithm is used in both series of simulations. It proposes a process delivering $2 \times k$ orders with price limits and quantities.

```
size=3 ;
for (int k = 1; k ¡= size; k++) do
    new LimitOrder(ASK, $2^{(k-1)}$, $10 - (k-1)$)
end
for (int k = 1; k ¡= size; k++) do
    new LimitOrder(BID, $2^{(size-k)}$, $12 - (k-1)$)
end
```

Algorithm 4: Automatic Generation of Orders

The complexity of the task is exponential and depends upon the number of orders the broker must deal with. When $k = 3$, one gets 720 potential permutations using the 6 orders (6!). This number of permutation is raised to 40320 when $k = 4$. On a 4-cores computer, this calculation takes around 1.5 seconds, which implies that a heuristic should be found to approximate

Table 3 broker order set O; orders interpretation : {id, direction, price, quantity}

$k = 3$	{A, Ask, 10, 1},{B, Ask, 9, 2}, {C, Ask, 8, 4}, {D, Bid, 12, 4}, {E, Bid, 11, 2}, {F, Bid, 10, 1}
$k = 4$	{A, Ask, 11, 1}, {B, Ask, 10, 2}, {C, Ask, 9, 4}, {D, Ask, 8, 8}, {E, Bid, 11, 8}, {F, Bid, 12, 4}, {G, Bid, 13, 2}, {H, Bid, 14, 1}

the optimal solution in an acceptable time. Notice for example that with the 4-cores computer, 12 pending orders necessitate nearly 5 hours of computing time; grid computing appears therefore necessary to get a solution at short notice, even if it cannot solve the problem. A realist number of pending orders at the broker's desk – for example 20 – should be arranged optimally in a few seconds, which for the moment, cannot be done. This necessitates to develop an appropriate heuristic. We report the results of this set of experiments in Table 4.

Table 4 Results, *systematic internalization*

n=3, 6 orders

	Num of Sol.	Card. of $\aleph$	U	Example of $O*$	w_B	Max w_B $^{(1)}$
Utilitarian	720	720	0	$\{A, B, C, D, E, F\}$	160	160*
Elitist	40	720	12	$\{A, B, E, D, C, F\}$	160	160*
Egalitarian	56	720	-2	$\{A, B, D, E, F, C\}$	160	160*

* $\rightarrow$ sequence in $\aleph$ to obtain this result : $\{A, F, D, B, E, C\}$

n=4, 8 orders

	Num of Sol.	Card. of $\aleph$	U	Example of $O*$	w_B	Max w_B $^{(1)}$
Utilitarian	40320	40320	0	$\{A, B, C, D, E, F, G, H\}$	320	338**
Elitist	288	40320	35	$\{A, B, C, F, G, E, D, H\}$	320	338**
Egalitarian	152	40320	-4	$\{A, B, C, F, G, D, E, H\}$	320	338**

** $\rightarrow$ sequence in $\aleph$ to obtain this result :$\{A, G, B, H, E, C, F, D\}$

$^{(1)}$: Maximum possible welfare for the broker

The left-hand side of the table (column 1 to 4) confirms that different social welfare measures deliver different optimal sequencing O^* for the order set generated by our algorithm (see Algorithm 4). These results clearly illustrate another important issue raised in this research : the broker own interests do not match his clients ones. For example, in the case where 8 orders must be sorted, $\{A, G, B, H, E, C, F, D\}$ is the more interesting sequence for the broker($w_B = 338$). With regard to the individual welfares of his clients, it is never an optimal solution (Utilitarian = 0, Elitist = 4, Egalitarian = -8). In other terms, whatever the social welfare measure, none is compatible with the broker own interests. This result points out a potential conflict of interests between these categories of economic agents (investors and intermediaries), which requires at least some supervision, and probably some regulation.

Conclusion

In stock markets, brokers hold a central position where they have the possibility to influence price dynamics in ordering the flow of orders they receive from their customers. Even if their activity is strictly monitored by professional association, they ultimately could arrange the pending orders from their clients so to capture maximum benefits for themselves (a behavior called "front running" in finance) or, alternatively, arrange these orders in sequences granting fairness amongst their clients. We therefore show in this paper, that since the broker's own interests do not match his clients' ones, order sequencing should be carefully set *before* posting the orders to the market so to match a predefined "fairness among clients" rule. We also defend that only an agent-based decision support system can solve this highly complex task. For the moment, we only propose an algorithm that never scales to find the best orders sequence execution. Nevertheless, the finding of heuristics with appropriate features and lower complexity is mandatory for future extensions.

References

1. K. J. Arrow. *Social Choice and Individual Values.* Yale University Press, 2nd edition, 1951.
2. C. Comerton-Forde, A. Frino, C. Fernandez, and T. Oetomo. How broker ability affects institutional trading costs. *Accounting and Finance*, 45(3):351–374, 2005.
3. R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223–236, 2001.
4. L. Harris. *Trading & Exchanges.* Oxford University Press, 2003. Chapter 11, "Orders anticipators".
5. P. Mathieu and O. Brandouy. A generic architecture for realistic simulations of complex financial dynamics. In Y. Demazeau, F. Dignum, J. Corchado, and J. Perez, editors, *Advances in Practical Applications of Agents and Multi-Agents Systems*, volume 70 of *Advances i intelligent and soft computing*, pages 185–198. Springer, 2010.
6. H. Moulin. *Fair Division and Collective Welfare.* MIT Press, 2004.
7. J. v. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 3rd edition, 1953.
8. Y. Nevmyvaka, M. Kearns, A. Papandreou, and K. Sycara. Electronic trading in order-driven markets: Efficient execution. pages 190–197, 2005.
9. A. Nongaillard, P. Mathieu, and P. Everaere. Nash welfare allocation problems: Concrete issues. In *Intelligent Agent Technology (IAT'10)*, pages 32–39, 2010.
10. J. Rawls. *A Theory of Justice.* 1971.