Portfolio Optimization under Market Impact Costs

Christian Oesch
Faculty of Business and Economics
University of Basel, Switzerland
Email: Christian.Oesch@unibs.ch

Dietmar Maringer
Faculty of Business and Economics
University of Basel, Switzerland
Email: Dietmar.Maringer@Unibas.ch

Abstract—This study presents a methodology for evolving mean-variance efficient portfolios when the agent is facing market impact costs. We use Grammatical Evolution, a form of Genetic Programming, to create portfolio strategies on an artificial market suited to simulate market impact. Classical portfolio selection as introduced by Markowitz is a well-established method to select securities based on their underlying returns and variances. This framework works well in an idealized world, where there are no market frictions and the true returns of the assets are known and normally distributed. In the real world however we face a range of problems such as transaction costs. For an active portfolio manager, transaction costs can consume a substantial amount of information value (also called a manager’s alpha). One part of the transaction costs which are implicit rather than explicit are market impact costs. There has been extensive research which looks at the problem of building or liquefying a given position when facing market impact costs but it might be beneficial to look at the problem from a broader perspective where the decision which assets to include into the portfolio has not been made. We find that on the artificial market, Grammatical Evolution is able to construct portfolio strategies which considerably outperform a linearly built-up Markowitz tangency portfolio by limiting the invested amount and adjusting the portfolio weights.

I. INTRODUCTION

Classical portfolio selection as introduced by Markowitz [1] is a well-established method to select securities based on their underlying returns and variances. This framework works well in an idealized world, where there are no market frictions and the parameters of the returns of the assets are known and are normally distributed. In the real world however we face a range of problems. Transaction costs are one of the most important topics to consider. For an active portfolio manager, transaction costs can consume about 40% of the information value (also called a manager’s alpha) [2]. In the case of a passive management, transaction costs become important because the portfolio needs to be rebalanced based on its performance and updated return estimates.

One part of transaction costs which are implicit rather than explicit are market impact costs. Institutions with large funds often face the problem that through the sheer size of their asset positions they become a significant market force for specific securities and thus influence the prices of these titles. Since price reactions to market transactions of such an institution will usually be opposed to its motives, they face a cost dependent on the size of their trade; we say that they bear market impact costs.

This poses an interesting problem for the selection of assets, since not only return and variance are important, but also how easily it is to rebalance, liquefy or build up a desired position. Part of the problem includes a second risk-return trade-off, since the investor can either build up his position very slowly, resulting in smaller market impact costs but higher execution risk. The execution risk describes the possibility of a price movement in an unfavorable fashion. When building up his position quicker he will face higher costs but lower execution risk. So far only few studies have addressed this problem extensively in an integrated manner where the influence of market impact on the portfolio has been taken into account during the asset selection process (see for example [3]). There has been thorough research which looks at the problem of building or liquefying a given position when facing market impact costs ([4], [5] and others) but it might be beneficial to look at the problem from a earlier perspective when the decision which assets and by what amount to include has not yet been made. This has been suggested by several industry professionals but it seems that the academic world couldn’t yet address this problem.

There exist quite a few micro-structural models which try to explain market impact by asymmetric information, such as the early works by [6], [7] and [8] and newer studies such as [9] which explains the effects by slow-moving capital. However they don’t account for both the temporary and permanent effects seen in empirical studies. Studies which describe both temporal and permanent effects only focus on specifying the form of the market impact function without micro-structural foundations such as [10], [4] and [11]. These studies concentrate on finding optimal transaction execution trajectories given simple linear or nonlinear price impact functions. Another important body of research which goes one step further by enhancing the models with limit order book mimicking functions can be found in [12] and [13]. However, all these models can only go so far in replicating the dynamics of a real order book and we see agent-based modeling approaches as an important advance in the study of market impact. Based on the statistical theory of the continuous double auction [14] a number of zero-intelligence agent-based modeling approaches have been introduced. The most advanced of these models is a refinement [15]. We will use an extended version of this market by introducing a group of fundamental agents to make sure the Grammatical Evolution algorithm cannot influence the market price in an unreasonable fashion. In the next section we will take a closer look at this market, but the interested reader is advised to consult the original study [15] for a deeper
understanding of the setting.

This paper presents a methodology for evolving mean-variance efficient portfolios when an agent is facing market impact costs. We use Grammatical Evolution to create portfolio strategies on an artificial market suited to simulate market impact.

II. The Artificial Market

The artificial market is an extended version of the proposed agent-based modeling approach to study price impact by Cui and Brabazon [15]. This market mimics a modern electronic double auction limit order book as used by most equity markets. The authors of the study find that their zero-intelligence agents produce market impact functions in relative order sizes which are too large when compared to the real world. However, we will still employ this model as it is the best approach available so far. Additionally we introduced fundamental agents which observe the fundamental value of the security and base their decision whether to buy or sell on the information of the asset’s underlying price. Before the actual market simulation is run, the fundamental price process for each security is generated from a correlated geometric Brownian motion:

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where $W_t$ is a correlated Wiener process, $\mu$ is the drift, and $\sigma$ is the asset’s constant volatility. This price process is revealed periodically to the fundamental agents. Now our artificial market consists of a number of zero-intelligence agents (ZIAgents), a number of fundamental agents (FAgents) as well as one mean-variance optimizing Grammatical Evolution agent (GEAgent). All agents have access to a central limit order book where they can trade a total of two different securities. Holding cash does not pay any dividends but acts as a riskless alternative which does not cause any transaction costs. Trading is modeled sequentially. In every period for each security a FAgent or ZIAgent is chosen randomly and gets the option to place or cancel an order. Every one hundredth period is reserved for the Grammatical Evolution agent to place his order.

The zero-intelligence agents randomly decide to buy or sell in each period with equal probability. Once they have picked their side of the trade, they randomly decide to place or cancel a limit order or to submit a market order.

The fundamental agents are in most respects the same as ZIAgents except for how they choose their trade position: given the market price is higher than the fundamental price, they will choose to place a sell order and if the market price is lower, they will choose to place a buy order.

The Grammatical Evolution agent is the main subject of the study. He is only allowed to place market orders and uses the signal from the Grammatical Evolution algorithm to decide whether to buy or sell and which amount of securities to trade. The details of the Grammatical Evolution setup are discussed in the next section.

Once an agent has been selected, the way ZIAgents and FAgents place their order (except for the buy and sell decision) are equal. With a probability of $\lambda_n$ they will do nothing, with probability $\lambda_m$ they will submit a market order, with probability $\lambda_t$ they submit a limit order, and with probability $\lambda_c$ they cancel a previously entered limit order. If they decide to place a limit order they will have four further options. With probability $\lambda_{in}$ they submit a crossing limit order which will be placed at the opposing price; a buy (sell) order will be placed at the best ask (bid) and will lead to immediate execution (at least partially). With probability $\lambda_{respr}$ they place their limit order uniformly between the bid and ask prices, with probability $\lambda_{spr}$ they place their buy (sell) order at the best bid (ask) and with probability $\lambda_{offspr}$ they place their buy (sell) limit order at a lower (higher) price than the best bid (ask) price in the order book. Order sizes are generated from a log normal distribution and relative off-spread limit prices are drawn from a power-law distribution. This seems to be consistent with empirical research [16], [17] and [18].

Whenever a new order arrives, the order book matches any possible trades and bid and ask prices are set. The matching happens after the rules of an electronic double auction limit order book. Each order book has two sides, one side lists the buy (bid) orders descending in price and the other side lists the sell (ask) orders in ascending order. The difference of the first entry in price between the two sides is called the bid-ask spread. Whenever a new order arrives which leads to a bid-ask spread smaller or equal to zero a trade occurs. When a trade takes place the corresponding orders are matched. If an order is only partially filled by volume, it usually stays in the order book until its volume is consumed by an opposing order. A more in-depth description of the workings of a double auction limit order book can be found in [14].

III. Grammatical Evolution

Grammatical Evolution (GE), first introduced by [19], is a grammar-based form of Genetic Programming. It is a population-based search mechanism which uses evolutionary or other heuristic algorithms to evolve a set of possible symbolic solutions to a given problem. By combining fundamental rules of biology and formal grammar definitions, runnable computer programs can automatically be built and executed without human intervention. As described in [20], [21], Grammatical Evolution uses a string of pseudo random numbers to represent different genes on a chromosome, called genome. Following a specific Backus-Naur-Form (BNF) grammar definition, a set of rules can be derived from the integers on the genome which leads to an executable program, the so called phenotype. One of the advantages over other Genetic Programming systems lies in the closure of every rule set which is evolved. The concept of closure, as defined by [22], requires that for every genetic operation, such as crossover or mutation, a valid function set is defined. Otherwise the program becomes uncompileable, rendering the individual invalid.

Grammatical Evolution is heavily inspired by the biological process of generating proteins from the genetic code of an organism [21]. In nature the genetic information is mostly encoded in the DNA (deoxyribonucleic acid) of the organism.
The DNA can be viewed as a string sequence of four different acid pairs, called nucleotides, which are used to encode the information to build amino acids. To produce the proteins, the DNA is first transcribed into a different format, the RNA. A group of three nucleotides, called a codon, is used to specify an amino acid. Once put together, the amino acids form a protein. This protein leads together with environmental factors to a phenotypic effect such as eye color or height of an individual [21]. Figure 1 shows a schematic comparison of the natural process of creating proteins.

The mathematical operators are the fail-safe versions as usually used in Genetic Programming. This mapping is used in the study to generate buy and sell signals for a total of 2 securities based on a set of periodically estimated input parameters. The rolling window estimated inputs from the artificial market are based on a set of periodically estimated input parameters. The experimental run starts with a warm-up phase where the order size. If the signal is positive the agent is to buy or sell the asset considered and also adjusts the signal serves as a decision variable whether the agent cannot go short in either cash or assets. After these adjustments the signal describes the security’s current proportional position in the agent’s portfolio. The output signal is scaled by the total amount the GEAgent can purchase with his total wealth and is capped such that the agent cannot go short in either cash or assets. After these adjustments the signal serves as a decision variable whether the GEAgent is to buy or sell the asset considered and also implies the order size. If the signal is positive the agent is to buy the asset and if it is negative the agent is to sell the asset. The absolute value of the signal rounded to the next integer implies the order size.

IV. Experimental Setup

The experimental run starts with a warm-up phase where only ZIAgents and FAgents are allowed to trade for 50’000 trading periods to get rid of any possible starting effects. In the second phase, the GEAgent starts with an empty portfolio and an amount c in cash and is allowed to place orders every
100th period. At the end, again the market is only open for ZI- and FAgents for 25’000 periods to ensure that any excessive temporary market impact the GEAgent has caused can be reversed.

Figure 2 shows the interface between the artificial market and the Grammatical Evolution’s individuals. Each individual of the population acts as one artificial market’s GEAgent and trades based on its computed signals for 50’000 periods. This procedure is repeated for 10 times with a set of random seeds which remain constant for the different generations. The fitness function sums the Sharpe-Ratios of the ten differently seeded evaluations:

\[
    \text{fitness} = \sum_{i=1}^{10} \frac{r_i}{\sigma_i}, \tag{2}
\]

where \( r_i \), the average excess return on each evaluation, divided by its standard deviation \( \sigma_i \), is the Sharpe-Ratio since the risk-free return is zero. In the next generation, the fitness value will decide if the individual survives or is replaced by a new individual. New individuals are produced by the genetic operations of crossover and mutation. This process is repeated for 75 generations and yields the final individuals which are the algorithm’s solutions to the portfolio optimization problem. The settings for the GE algorithm and the evolutionary settings are presented in Table I. We use a genome string with 128 integers and allow the algorithm to create phenotypes up to a depth of 16 steps. This depth limitation was rarely binding in our experiments. The population size for each experiment has been set to 100 individuals. We use a simple truncation selection to find the parents for the next generation. While a lot of other studies use a roulette-wheel based selection process, in this experiment we haven’t found any advantages of using a more sophisticated solution. The crossover probability is quite high due to the long genome.

Table II shows the artificial market settings. All parameters have been selected by analysing previous experiments on the problem. The two securities which can be traded are identical in all respects except for the trading probability which is halved for security B. This leads to a thinner order book as well as a less smooth price movement. The event type probabilities have been selected to have the same proportional relationship as in [15] but have been amped up, so the system requires less periods to build up the order book and to make the price process smoother. Limit order type probabilities, order size parameters and the relative off-spread price parameters have been taken directly from [15]. This setup theoretically can be estimated from real order book information by maximum likelihood except for the percentage of fundamental traders on the market. This is why we have conducted experiments with different agent proportions.

Table III lists the different settings for the starting cash of the GEAgent as well as the percentage of FAgents on
the market. The percentage values in brackets indicate the percentage of market volume the GEAgent creates when he equally invests all his cash in asset A and B. For each setting, 21 runs have been performed. To compare the results, in each run a benchmark strategy is evaluated. The benchmark strategy is to linearly build up a tangency portfolio using all the available resources of the agent and then holding it for the duration of the cool-down period. This corresponds to the case when the portfolio manager does not consider the effects of market impact on the portfolio composition. The linear splitting of large orders is the industry standard to minimize execution risk while maintaining low market impact costs.

To conclude the experiments, the strategies found by GE are evaluated on a new set of post-sample seeds. Again these evaluations are performed ten times and the fitness scores are evaluated on a new set of post-sample seeds. Again these settings it might be beneficial to let the experiment run for a few more generations to achieve even better results. These values in brackets indicate the percentage of market volume the agent consumes a lot of the returns earned by holding the assets. In these settings it might be beneficial to let the experiment run for a few more generations to achieve even better results.

All changes in positions that occur later than period 50’000 are due to remaining (partially filled) orders in the order book. There doesn’t seem to be much difference between the less frequently traded asset B and the more frequently traded asset A. This is surprising, since asset B is only traded half as much as asset A. The solutions by GE don’t seem to differ much from the tangency solution. Nevertheless compared to the benchmark, especially when the cash reserves are big, it seems to be optimal to only invest part of the total fund into the securities. When one invests too much in an asset the cost induced by market impact becomes too large and consumes the possible gains from holding the asset.

In table IV the post-sample performances are listed and compared to the results of the benchmark strategy. Especially in settings where there are fewer fundamental agents GE considerably outperforms the benchmark strategy. This might be caused by the fact that the algorithm learns to exploit its own market power even for longer durations. In setting 4 and 5 when the invested amount becomes very large, GE is able to find better solutions than the benchmark by investing less cash into assets and thus is able to avoid the market impact costs which are higher than the additional gain from holding the larger position.

 VI. Conclusion

In this study we have expanded on a zero-intelligence agent-based modelling approach by introducing a group of fundamental agents. This leads to a more stable market and lets us introduce a Grammatical Evolution agent to test its performance for the task of optimizing large portfolios on an illiquid market. The results of this experiment suggest that Grammatical Evolution is a useful tool when optimizing large portfolios. The grammar used is still very open, and we assume that the performance could be enhanced even further by specifying a more tailored grammar. However before doing this, we would like to create a more sophisticated agent-based market. To make a more meaningful comparison of the benchmark strategy to the solutions found by GE, we should build an artificial market which captures the most important properties of market impact more closely. As can be seen in newer literature about market impact, e.g. [24], strategic considerations by the agents lead to important properties in

<table>
<thead>
<tr>
<th>Settings</th>
<th>Setting 01</th>
<th>Setting 02</th>
<th>Setting 03</th>
<th>Setting 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cash c</td>
<td>1 BN</td>
<td>1 BN</td>
<td>1 BN</td>
<td>4 BN</td>
</tr>
<tr>
<td>(≈18.8%)</td>
<td>(≈18.8%)</td>
<td>(≈18.8%)</td>
<td>(≈75.2%)</td>
<td></td>
</tr>
<tr>
<td>FAgents</td>
<td>20%</td>
<td>10%</td>
<td>0%</td>
<td>5%</td>
</tr>
<tr>
<td>Settings</td>
<td>Setting 05</td>
<td>Setting 06</td>
<td>Setting 07</td>
<td>Setting 08</td>
</tr>
<tr>
<td>Cash c</td>
<td>2 BN</td>
<td>1 BN</td>
<td>300 MM</td>
<td>250 MM</td>
</tr>
<tr>
<td>(≈37.6%)</td>
<td>(≈18.8%)</td>
<td>(≈9.4%)</td>
<td>(≈4.7%)</td>
<td></td>
</tr>
<tr>
<td>FAgents</td>
<td>5%</td>
<td>5%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>Settings</td>
<td>Setting 01</td>
<td>Setting 02</td>
<td>Setting 03</td>
<td>Setting 04</td>
</tr>
<tr>
<td>GE (best)</td>
<td>192.82</td>
<td>248.89</td>
<td>825.45</td>
<td>814.42</td>
</tr>
<tr>
<td>GE (avg. 5 best)</td>
<td>181.72</td>
<td>214.46</td>
<td>591.77</td>
<td>636.89</td>
</tr>
<tr>
<td>GE (avg. all)</td>
<td>142.48</td>
<td>161.11</td>
<td>408.24</td>
<td>320.48</td>
</tr>
<tr>
<td>avg. Benchmark</td>
<td>340.45</td>
<td>248.66</td>
<td>121.58</td>
<td>181.05</td>
</tr>
<tr>
<td>Settings</td>
<td>Setting 01</td>
<td>Setting 02</td>
<td>Setting 03</td>
<td>Setting 04</td>
</tr>
<tr>
<td>GE (best)</td>
<td>328.71</td>
<td>291.13</td>
<td>359.51</td>
<td>339.18</td>
</tr>
<tr>
<td>GE (avg. 5 best)</td>
<td>302.30</td>
<td>280.17</td>
<td>284.87</td>
<td>304.66</td>
</tr>
<tr>
<td>GE (avg. all)</td>
<td>238.86</td>
<td>203.18</td>
<td>222.66</td>
<td>235.16</td>
</tr>
<tr>
<td>avg. Benchmark</td>
<td>60.97</td>
<td>248.07</td>
<td>268.03</td>
<td>229.02</td>
</tr>
</tbody>
</table>
Fig. 3. The series display the mean fitness evolution of all the elitist groups for the different settings.

Fig. 4. The position build-up trajectories for securities A and B. Period 0 marks the time after the initial warm-up phase, and period 50’000 marks the beginning of the cool-down phase.
the market impact functions and should be incorporated into a future version of this work. By introducing block orders which are split up into smaller orders and thus create correlated order flow in the market, we might take a promising step to unify the micro-structural models with the empirical facts observed on financial markets.

A second branch of research this study suggests is the specific task of optimally rebalancing a portfolio when an agent faces market impact costs, which we have not performed in this study because it is very unclear from what starting position one should start. The results however suggest that the tangency portfolio should be a good start since the basic composition of the portfolio found by GE very closely resembles this analytical solution.

REFERENCES