UNIVERSITÀ DEGLI STUDI DI TORINO
SCHOOL OF MANAGEMENT AND ECONOMICS

SIMULATION MODELS FOR ECONOMICS

Final Report

“ Stop-Loss Strategy ”

Prof. Pietro Terna
Edited by Luca Di Salvo, Giorgio Melon,
Luca Pischedda

Summary

A Stop-Loss Strategy in NetLogo

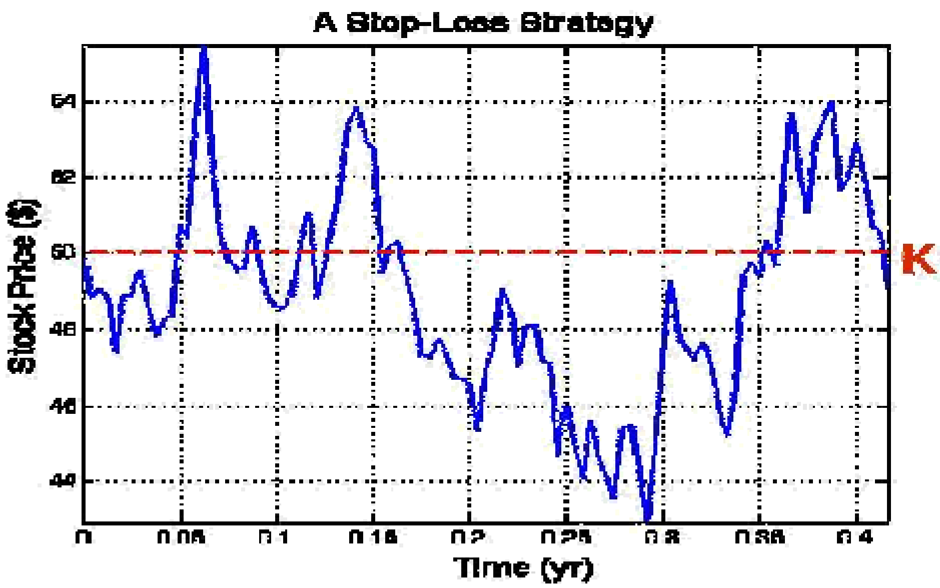
1. Introduction

The Stop-Loss strategy is a strategy involving the shortage of a call and the trading of a stock. While the former is an operation made once, the latter could require more trades depending on the path of the underlying.

Before explaining the strategy in detail, a distinction between the two possible positions taken by the investor is needed.
NAKED position: The investor is short a call. It produces a profit equal to the call price in t = 0 if the call is OTM at the expiry date, but leads to significant losses when the call expires ITM.

COVERED position: The investor is short a call and long the underlying (assumed to be bought at K, the strike price). It produces a profit equal to the call price in t = 0 if the call expires ITM, but leads to significant losses when it expires OTM.

The strategy kernel lies in the intersection between the two position.



This simple hedging strategy is to ensure that at time T, the bank owns the stock if the option closes in the money and does not own it if the option closes out of money.

Assumptions:

- Lognormality of asset values.
- Call priced using BS.

In t = 0 the investor sells a call and has an inflow equal to the call price. Since in t = 0 the spot price is lower than the strike price the stop-loss investor is only short a call. As soon as the stock price touches the strike price barrier from the down the investor goes long on the underlying, whose value is K. If the stock price hits again the barrier (this time from the up) the trader sells the asset, remaining naked on the call. This procedure is repeated every time the stock price equals K.

Therefore, the strategy is a combination of naked and covered positions that exploits only the advantages of the two. In fact, when the option is ITM the loss is avoided because the seller of the call holds the stock in the portfolio and the call's payoff(from the seller point of view) becomes Max (K – K, 0). On the other hand, when the option is OTM is not exercised, hence the payoff is 0. In any case, the stop loss investor obtains a payoff equal to the call price received in t = 0.

1.1 The Black-Scholes Model

One of the first issue to solve involves the pricing of the call. Given the above mentioned market, we implemented the Black-Scholes model to price the call.

The Black-Scholes model for calculating the premium of an option was introduced in 1973 in a paper entitled, "The Pricing of Options and Corporate Liabilities" published in the Journal of Political Economy. The formula, developed by three economists – Fischer Black, Myron Scholes and Robert Merton – is perhaps the world's most well-known options pricing model. Black passed away two years before Scholes and Merton were awarded the 1997 Nobel Prize in Economics for their work in finding a new method to determine the value of derivatives (the Nobel Prize is not given posthumously; however, the Nobel committee acknowledged Black's role in the Black-Scholes model).

The Black-Scholes model is used to calculate the theoretical price of European put and call options, ignoring any dividends paid during the option's lifetime. While the original Black-Scholes model did not take into consideration the effects of dividends paid during the life of the option, the model can be adapted to account for dividends by determining the ex-dividend date value of the underlying stock.

The model makes certain assumptions, including:

- The options are European and can only be exercised at expiration
- No dividends are paid out during the life of the option
- Efficient markets (i.e., market movements cannot be predicted)
- No commissions
- The risk-free rate and volatility of the underlying are known and constant
- Follows a lognormal distribution; that is, returns on the underlying are normally distributed.

The formula takes the following variables into consideration:

- Current underlying price
- Options strike price
- Time until expiration
- Implied volatility
- Risk-free interest rates

$$\boldsymbol{C} = SN(d_1) - N(d_2)Ke^{-rt}$$

C = Call premium
S = Current stock price
t = Time until option exercise
K = Option striking price
r = Risk-free interest rate
N = Cumulative standard normal distribution
e = Exponential term

s = St. Deviation
ln = Natural Log

$$\boldsymbol{d_1} = \frac{\ln\left({}^{S}/_{K}\right) + \left(r + {}^{s^2}/_{2}\right)t}{\sqrt[s]{t}}$$

$$\boldsymbol{d_2} = d_2 - \sqrt[s]{t}$$

The Black-Scholes pricing formula for call options.

The model is essentially divided into two parts: the first part, SN(d1), multiplies the price by the change in the call premium in relation to a change in the underlying price. This part of the formula shows the expected benefit of purchasing the underlying outright. The second part, N(d2)Ke^(-rt), provides the current value of paying the exercise price upon expiration (remember, the Black-Scholes model applies to European options that are exercisable only on expiration day). The value of the option is calculated by taking the difference between the two parts, as shown in the equation.

2. The model for the stock

In order to simulate the stock path we used an existing model called *g1_CDA_basic_model*.
It creates a number of agents that is arbitrarily chosen by the user. During the bargain phase, each of them has a 50% probability of being either a buyer or a seller. Plus, each agent is provided with a probability to pass and, therefore, not to conclude the transaction. Also in this case, we can make it more likely by moving the appropriate slider.
The procedure continues by giving to each agent a semi random price, whose components are a fixed amount increased or decreased by a random amount. Those prices are then ordered creating a vector whose components are increasing (for prices generated by sellers) and decreasing (for prices generated by buyers). The first element of this vector is the market price.
This procedure is continuously repeated eliminating the previous "first price" so that a new equilibrium price is created and a variation is always observed in the market. This will be useful when calculating the implied volatility and ensure a positive value of it.

2.1 Random Strategy

We wanted to compare the stop loss strategy with a winning random strategy. In order to make the random strategy winning we made random investors start with a long position in stocks. Plus, we gave random agents the possibility to purchase and sell at the same time, creating an asimmetry in the market.

The comparison with a pure random strategy would not have been significant since the stop-loss strategy tends to gain by costruction

After determining the procedure for the creation of prices, we have the inputs to set up a random strategy on the basis of bid/ask prices. When a bid price is created random agents sell. Viceversa, when an ask price is created random agents buy.

They all start with a long position and then in the next period they take a position opposite to the one taken in the previous period. This mechanism is regulated by the variable *randombuy* and *randomsell* can either take value 0 or 1.

```
ask randomAgents [
          if randomBuy >= 1 and random-float 1 < .5
                    [set randomPtf randomPtf + exeprice
                     set randomSell randomSell + 1
                     set randomBuy randomBuy - 1]]

ask randomAgents [
          if randomSell >= 1 and random-float 1 < .5
                    [set randomPtf randomPtf - exeprice
                     set randomSell randomSell - 1
                     set randomBuy randomBuy + 1]]
```

3. Evaluation of Stop Loss Strategy Parameters

In order to properly set up the strategy we need to use all the parameters involved in the computation of the call price, according to the Black-Scholes formula. A crucial role is played by strike which creates a partition of the space changing the composition of the portfolios of Stop-Loss investors and the magnitude of the cash flows as soon as exeprice hits its barrier.

Another important parameter is T, since for the strategy to work properly we need to calibrate the correct timing. In fact, when the call expires the cash flow occurring can either be 0 or K.

In order to generate the volatility of the stock we have to consider the log-returns of the underlying, generated by:

```
set priceVector fput exePrice priceVector
    if length priceVector > 1
    [let tmp2 ln ( item 0 priceVector / item 1 priceVector )
    set logVector fput tmp2 logVector]
```

Each price generated is put in the pricevector in first position, that collects prices generated in different periods thanks to interaction among agents. Starting from this vector we will define the volatility of the log-returns, defined as the natural logarithm of the last and second to last price.

When the vector gets too big, we eliminate the non relevant components by:

```
if length logVector > 700
    [
      set logVector butlast logVector
      set sigma standard-deviation (logVector) * sqrt (36400)
     ]
```

The strategy duration is 700*T, where T can be arbitrarily chosen. The call price is computed, according to the Black-Scholes formula, as a function of sigma, T, r, S and K.

The strike price is formed on the basis of the last exeprice and a random float which takes values between -1 and 1 , multiplied by 200. The strike price is only computed at the beginning of the strategy and does not change until the call reaches its expiry.

```
if k = 700 * T [ ask SLAgents[
                    set strike exePrice + (random-float 2 - 1) * 200
set d1 (ln (exeprice / strike) + (risk-free + ((sigma ^ 2) / 2)) * T) / ( sigma * sqrt ( T))
                    set d2 d1 - sigma * sqrt (T)

                    let a d1 / (2 ^ 0.5)
                    let b d2 / (2 ^ 0.5)
                    set Nd1 1 - 0.5 * erfcc a
                    set Nd2 1 - 0.5 * erfcc b
                    set BScall exePrice * Nd1 - strike * exp( - risk-free * T ) * Nd2
```

Since NetLongo did not provide us with an existing function for the cumulative distribution function of the Gaussian distribution, we used a numerical approximation of it using the error function of the Normal distribution.

At the starting date, depending on the largeness of the spot price with respect to the strike price, we set either a naked or covered position.

```
ifelse exePrice < strike
      [set slPtf slPtf + BScall set naked true set covered false]
      [set slPtf slPtf + BScall - exePrice set covered true set naked false
                              set price exePrice set buy true set sell false]
```

When the strategy is implemented, the stop loss traders enter in the price formation mechanism if the *exeprice* is higher than the strike price. Otherwise, they will keep a naked position.

The variable *checkstrategy* allows us to change the monitoring of the strategy. A chooser has been implemented to select the hourly, weekly or monthly monitoring.

After 700 / *checkstrategy* prices generated, corresponding to the desidered check period, the spot price is checked to verify whether it has hit the barrier or not. If it has from the up, the stop loss investor takes a naked position. Viceversa, if it has from the down, the stop loss investor takes a covered position.

```
if j = 700 / checkStrategy  [ ask SLAgents  [
                                            if naked and exeprice > strike
                                           [   set covered true set naked false
                                            set slPtf slPtf - exePrice * (1 + TransactionCost)
                                             set price strike set buy true set sell false]



                                          if covered and exeprice < strike
                                         [    set naked true set covered false
                                            set slPtf slPtf + exeprice * (1 - TransactionCost)
                                            set price strike set sell true set buy false ]
                                                    ]
                                          set j 0
                                        ]
```

Note that also in this case agents affect the price formation mechanism when the spot price hits the barrier of the strike price.

The counter j implemented counts 700 prices formed, which correspond to a working week. This value comes from the following assumptions:

- 1 price every 3 minutes
- 7 hours per day
- 5 days per week

"Checkstrategy" allows us to modify the checking timing of the spot price with respect to the strike price (and consequently monitor whether or not the hedging needs to be implemented). It appears like a string, but actually has a numeric value in the code. It can be:

- 1 If we want to have the check weekly.
- 0,25 if we want to have the check monthly. In fact, under the assumption of 4 weeks per month, 700/0,25 is 2800 (i.e. the number of prices formed in a month).
- 35 if we want to have the check hourly. In this case we check it every 20 prices formed (i.e. a hour of a working day).

When the counter reaches 700/checkstrategy, the StopLoss trader check hedging need, the counter is set equal to zero and the mechanism starts again.

The final step of the procedure occurs in T-1, when the strategy terminates and the call expires. If the stop loss investor has a naked position, then there is not any cash flow since the option is not exercised. If the stop loss investor has a covered position, he benefits of a positive cash flow of K, that is the amount paid by the buyer of the call for getting the underlying at a price lower than the spot price observable in the market at that moment.

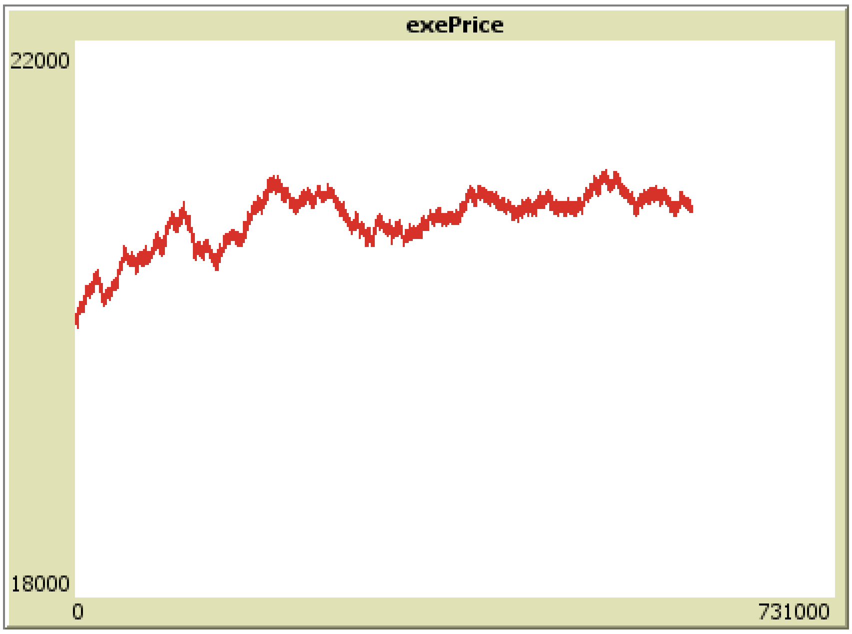
```
if k = 700 * (T - 1) and p > 0[ ask SLAgents[
                    if exePrice > strike
                     [set slPtf slPtf + strike* (1 - TransactionCost)  ]
                         ]
              ]
```

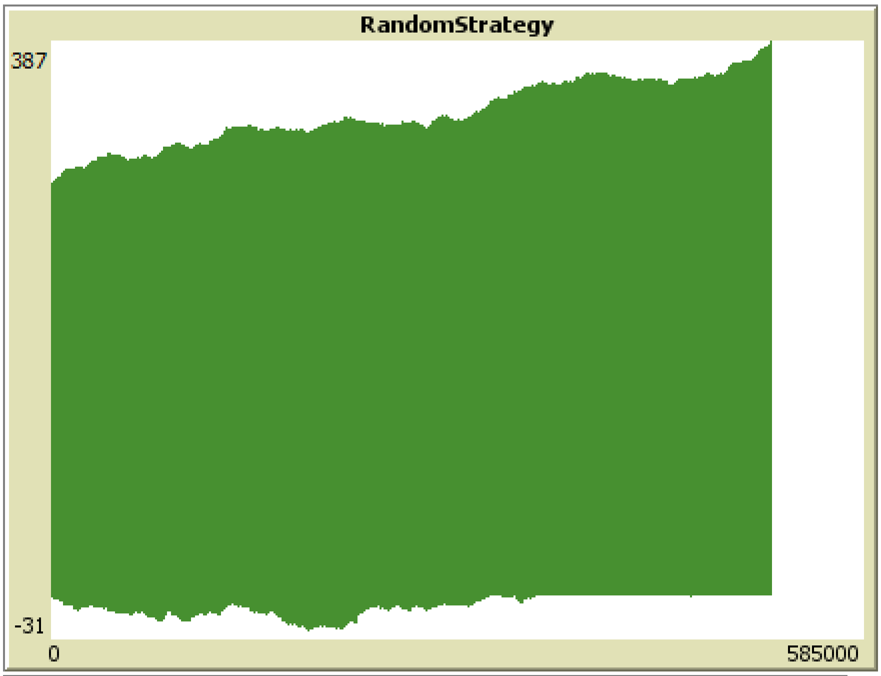
4. Simulation

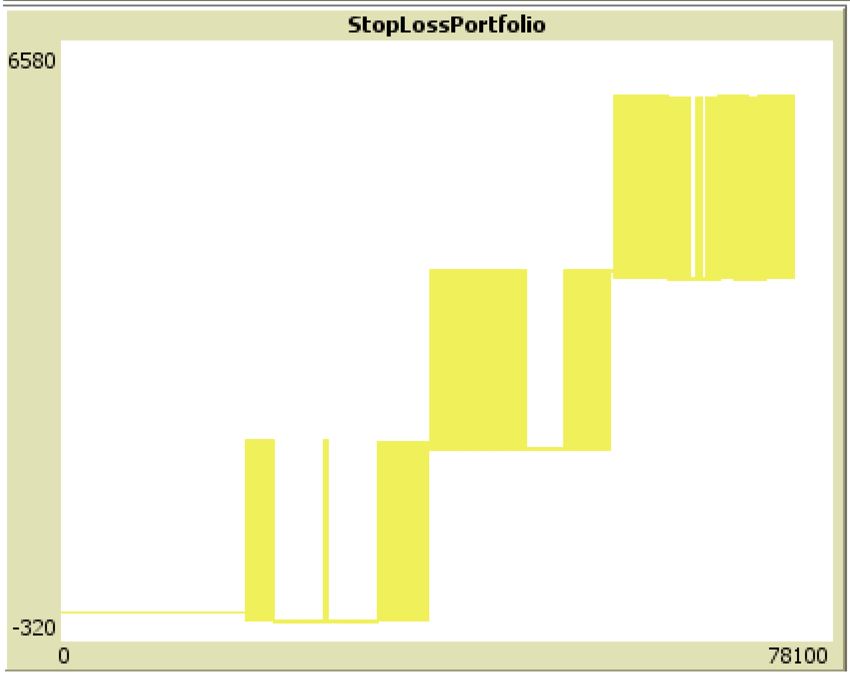
In this part we will deal with some empirical results obtained by running the program with different configuration of parameters. We decided to modify the values of the most crucial parameters, such as monitoring time, price shock and transaction costs. In fact, the strategy we built admits no losses (by that, the name "Stop-Loss") under the assumption of no transaction costs and stable environment.

nRandomAgents:70 *seed*:50 *nslAgents*: 10

Check period: 1 week
Price shock: Low (15)
Transaction costs: No







With a standard configuration of parameters, after performing the strategy for 4 years the graph highlights a gain. Any reduction in the portfolio of the stop-loss investor is due to the outflow deriving from the acquisition of the stock for hedging purposes.

The portfolio is checked every week to decide whether or not the investor needs to purchase/sell the stock according to the spot price observed.

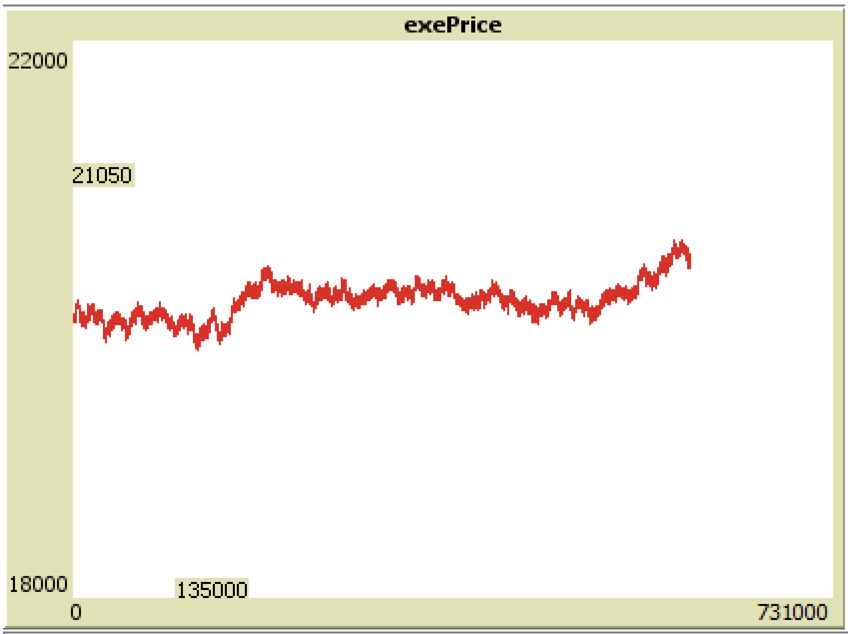
The low shock rate of the underlying makes hedging easier and ensures a good performance.

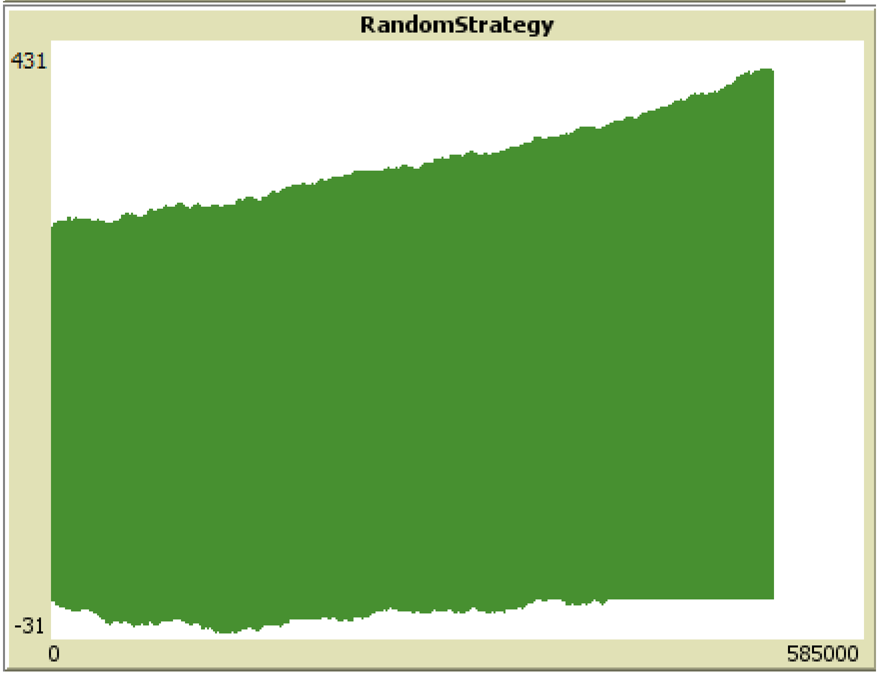
nRandomAgents:70 *seed*:40 *nslAgents*: 10

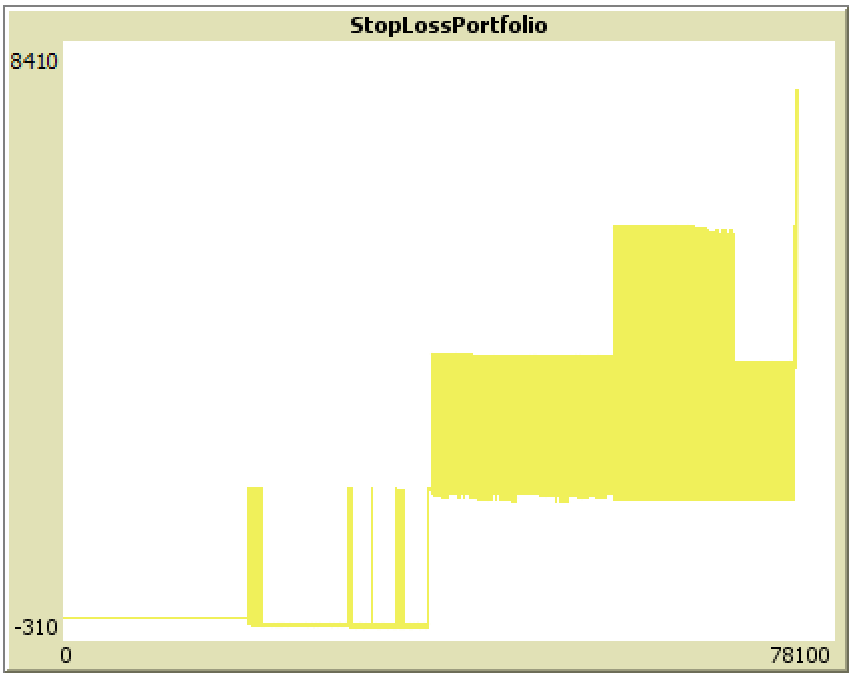
Check period: 1 hour

Price shock: Low (15)

Transaction costs: No







The pattern followed by the net cash flow of stop loss investors is very similar to the one of the experiment run by checking the price weekly. The absence of transaction costs, in fact, allows investors to make as many trades as they want and increases the efficiency of the strategy.

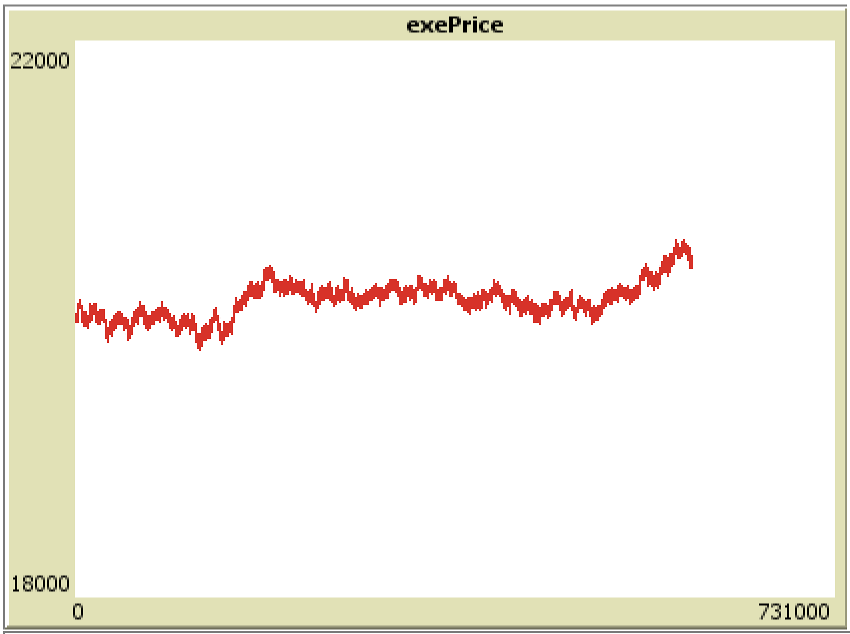
The hour strategy needs much more cash flows than the week strategy. In fact, checking the strategy more often implies more trades, as the chart highlights.

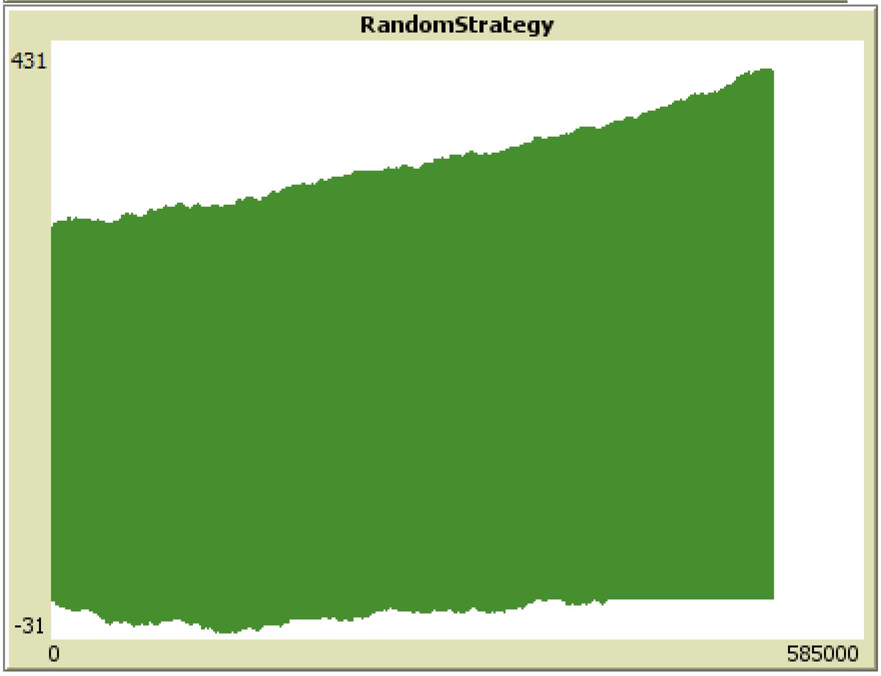
nRandomAgents:70   *seed*:30   *nslAgents*: 10

Check period: 1 hour

Price shock: Low (15)

Transaction costs: Yes (5%)



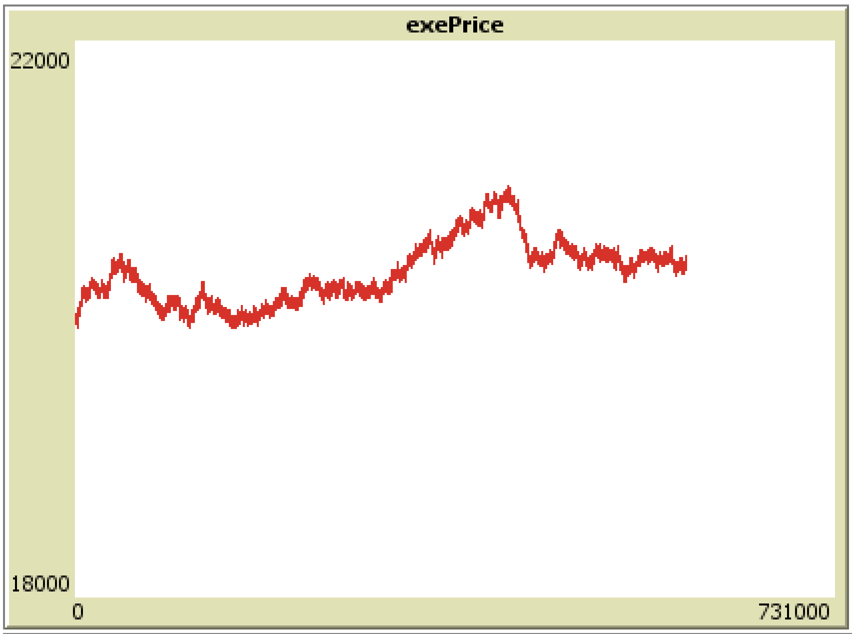


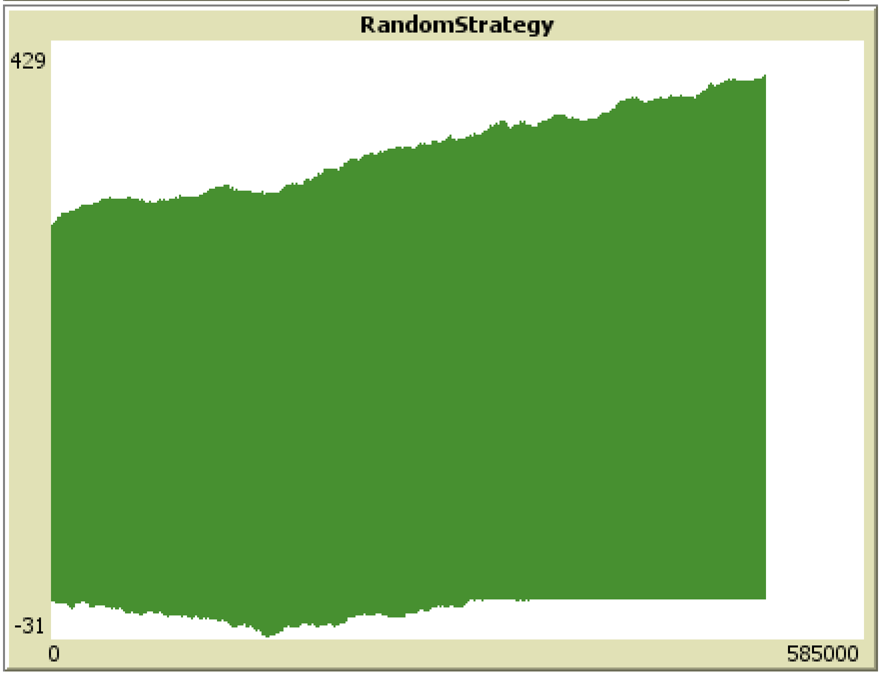
The graph highlights how the introduction of transaction costs negatively impacts on the net cash flow obtained by stop loss investors. In fact, since the maximum payoff obtainable by them is the price of the call this amount is immediately absorbed by the cost the trader must pay for purchasing/selling the stock any time it hits the barrier.

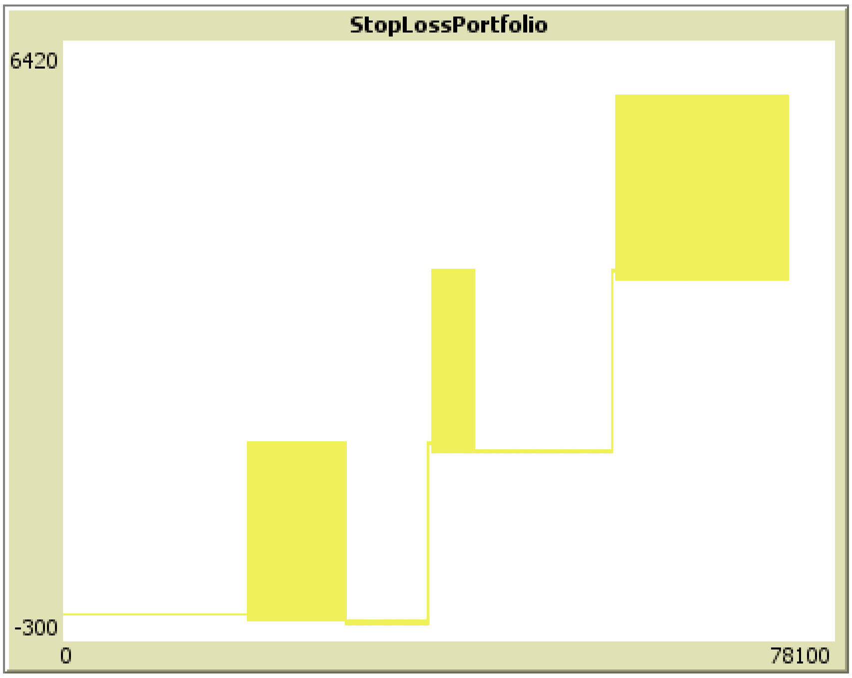
Check period: 1 month

Price shock: Low (15)

Transaction costs: No







With a standard configuration of parameters, after performing the strategy for 4 years the graph highlights a gain.

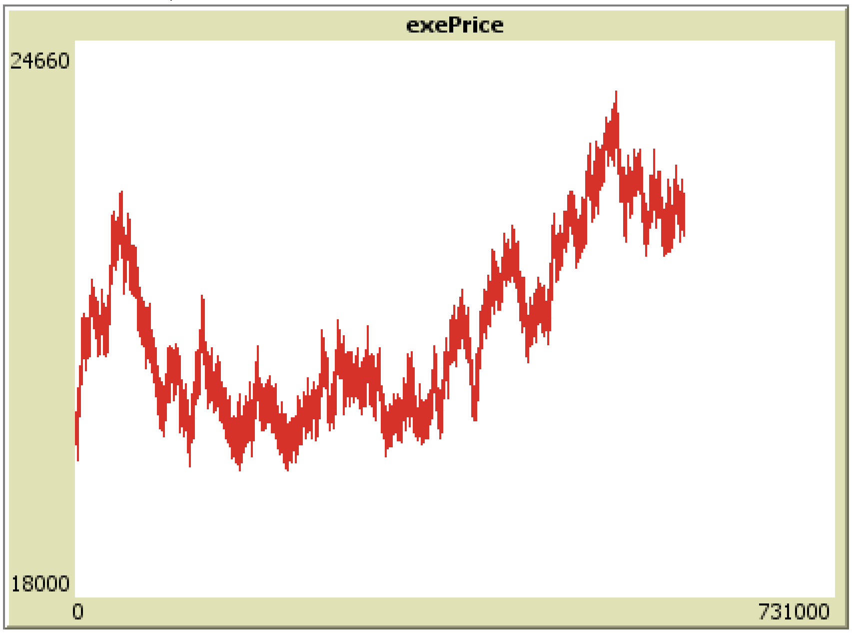
The portfolio is checked every month to decide whether or not the investor needs to purchase/sell the stock according to the spot price observed.

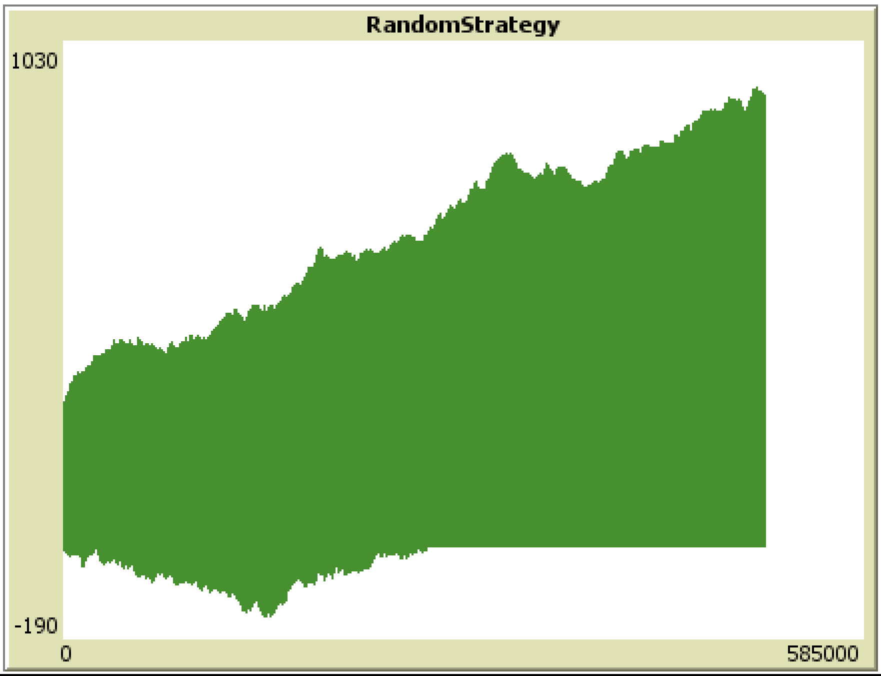
The shock rate does not impact significantly on the performance, but we can see a better gain with an higher shock rate.

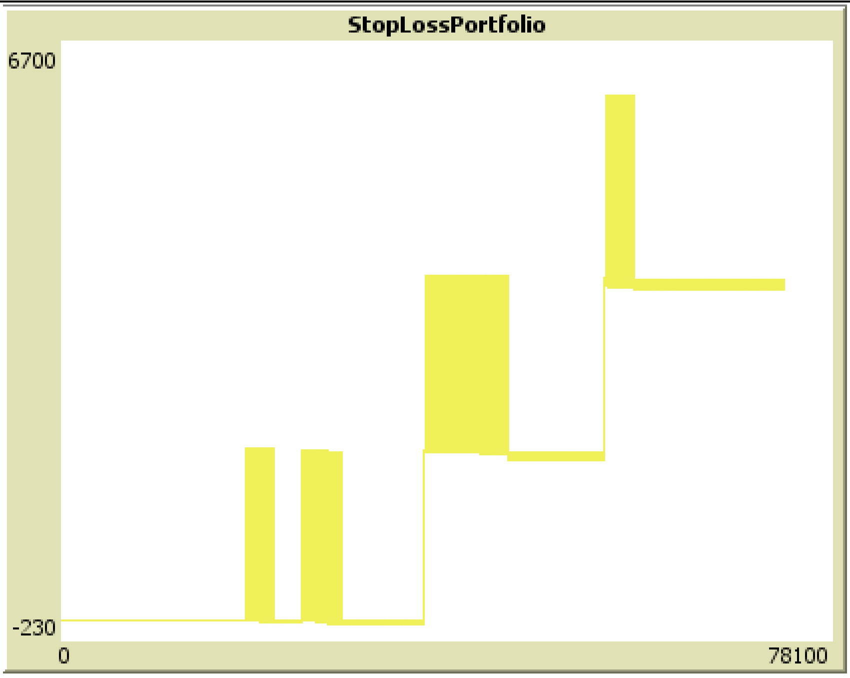
Check period: 1 month

Price shock: High (90)

Transaction costs: No







When we increase the shock, and consequently the implied volatility, the performance of the strategy worsens. In fact, the strategy works better in a stable environment where transactions are made exactly at K and monitoring is continuous. Enlarging shock and check period the trader is not able both to respond immediately to market movements and to control the path of the underlying, which becomes much more jagged.

5. Conclusions

Graphs show how the stop loss strategy, in a scenario without transaction costs, ensures a gain much higher than a random strategy. In fact, the stop loss is a perfecly covered strategy that admits no losses. On the other hand, the random strategy has no cover and corresponds to gambling in the financial market without any use of derivative instruments and it can lead to a gain but also to significant losses.

The empirical tests show how the stop loss strategy, in a standard scenario, ensures a gain much higher than a random strategy. In fact, the stop loss is a perfecly covered strategy that admits no losses. On the other hand, the random strategy has no cover and corresponds to gambling in the financial market without any use of derivative instruments and it can lead either to a gain or a significant loss.

Varying the configuration of parameters could compromise the outperformance of the strategy if we take the random strategy as a benchmark. In fact:
Adding transaction costs to the model leads to a loss when we increase the number of checks, while does not affect the performance significantly when the check is made monthly.
When we increase the shock (leading, therefore, to an increase of the volatility of the underlying) the gain gets lower due to a more unstable environment that makes trades less convenient. In fact, on average, purchases become more expensive faster.

That is why the strategy explained in these chapters is not applied in the real world. Even though the simple idea behind is fascinating, the disturbances and costs occuring in the real world make it less efficient than other hedging strategies.