Clinical Cancer Investigation Journal-ISSN: 2278-0513

ICS: A New Mathematical Formula (Algorithm) for Lot Size Calculation

Touraj Ostovari
Department of Computer Software Engineering from Tabriz Technical and Vocational University.
Toraj.ostovari@gmail.com

Abstract

Lot size calculation is one of the global trading world's current concerns that every trader in financial markets requires to resolve their situation somehow [1][2]. Still, most common methods cannot perform in all financial markets and encounter errors (i.e., they show an output of zero instead of the lot size output, which shows a lack of optimization). The current method performs all these calculations, but its risk management model requires high-accuracy trading and calculations of possible losses for the trader. This method is recommended for large banks and financial institutes with large assets or conducting Martingale transactions. However, this method can be used for non-Martingale transactions as well. The current study does not require any mathematical proofs because it only uses simple mathematical operations, and its results were tested in financial market environments with a sample output presented later on from the Meta Trader 4 software with a low drawdown rate and high efficiency. This is a completely novel formula for calculating trade lot size in Foreign Exchange (Forex) or other financial markets with lot-size-based trade. The results of this formula were calculated with Meta Trader in a real environment with real accounts. This paper presents no mathematical proofs; it only discusses the calculation method.

Keywords: *Risk Management, Lot Size Calculation, Financial Markets, Foreign Exchange Market, Capital Markets, Economy.*

Introduction

The current calculation model uses five different parameters (capital amount, fixed risk, floating risk, pip amount, and pip risk amount coefficient), which are rather the same as another common method that will be discussed later on, alongside nested loops and simple mathematical operations such as multiplication and division. This formula's coding and simple mathematical equations will be explained later on. Some disadvantages of this method are its complex and repetitive calculations for people during each transaction, which are rather simple for computer systems. In addition, this formula does not include the loss percentage of each pip, so this loss percentage must be calculated manually (if needed) for each transaction, or there could be an increase in the fixed risk value to reduce the percentage loss. An advantage of this formula is giving the risk-taking control to the trader because this formula consists of five input parameters that can determine the transaction risk-taking value and better control the calculation size. In other words, this formula contains more than the other common method that only has the risk-taking value as its input. An investment company with great drawdown results implemented all calculator results on live accounts. I am hoping that this article will be useful for human society.

Implementation (method):

To start, this formula has 5 initial variables:

- Initial Capital (n_equity)
- Floating Risk (ICS_FLOATRISK)
- Fixed Risk (ICS_FIXRISK)
- The Possible Pip Value in Loss (Pip amount)
- The Risk Value Coefficient of Each Pip in Loss

Increasing the fixed risk value and maintaining the floating risk value can reduce the transaction size (risk-taking) with a more diverse output volume than other single input variable common methods.

1. Formula Calculation - Step One:

Z = (((n_equity.ICS_FLOATRISK)/100).ICS_FIXRISK/Pip amount)

Thus far, the coding commands and mathematical operations are the same.

2. Formula Calculation - Step Two (Final):

We create a repeating loop with the Z value in temp. The repeated loop iterates i < ICS_FIXRISK times. Then, the temp value calculations that has gotten from Z after the loop finishes, or it shows the temp variable value with two decimal places to the user.

The loop in coding:

```
Float temp = Z;
for (int i = 0; i < ICS_FIXRISK; i++)
{
temp = temp / 2;
}
Print (“%.2f”, temp);
```

The repeating loop in mathematics[3][4][5][6]:

$$Z_n = \begin{cases} Z_{n+1} = Z_n mod2, & n < z \\ Z, & n = z \end{cases}$$

Calculation Sample and Comparision alongside Transaction Results with the Current Lot Size Calculation Method - Discussion:

The following is the formula used for most lot size calculations in the Forex or any other financial markets with lot-size-based trade:

Lot Size = ((Balance * Percentage) ÷ Pip Amount) ÷ 100,000

On Gold Market Example:

Lot 0.48 = ((100,000 * 0.5) ÷ 1.03) ÷ 100,000

Lot 0.09 = ((100,000 * 0.1) ÷ 1.03) ÷ 100,000

On NASDAQ Market Example:

Lot 0.00 = ((100,000 * 0.5) ÷ 65.81) ÷ 100,000

Lot 0.01 = ((100,000 * 1.0) ÷ 65.81) ÷ 100,000

As you can see, this formula receives some errors from time to time and returns a zero value while the user has less control over the input values for capital risk management.

Now, this is a sample from the calculation of this paper:

```
Float temp = (((100,000. 1) / 100). 14 / 65.81);
for (int i = 0; i < ICS_FIXRISK; i++)
{
temp = temp / 2;
}
Print (“%.2f”, temp);
```

Output in C programming language:

0.01 = Lot size

This formula returns the 0.01 lot size output, a sample of the NASDAQ index with the lowest possible risk-taking value. Increasing the floating risk and reducing the fixed risk can lead to higher risks (i.e., larger lot sizes).

The above example has a price difference of 658.10, which leads us to choose a low-risk value of 10 by dividing the price difference by 10, 100, and 1000 (low, medium, and high risk).

Here is another example from the cryptocurrency (BitCoin) market:

Input Price: 42568.9 Stop Loss Price: 30960.8

The prince difference between these two is 11608.1. We divide 11608.1 by 10 to 1000 based on the risk-taking value of this market, which results in a medium risk by dividing the price difference by 100:

11608.1/100=116.081

```
Float temp = (((100,000 . 5) / 100) . 14 / 116.081);
for (int i = 0; i < ICS_FIXRISK; i++)
{
temp = temp / 2;
}
Print (“%.2f”, temp);
```

Output in C programming language:

0.04 = Lot size

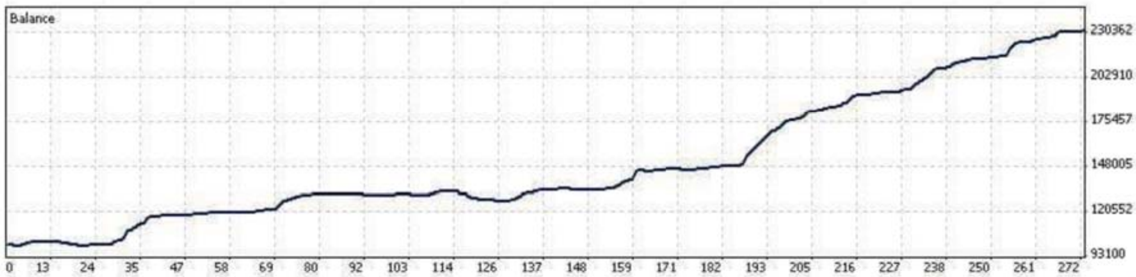
The pseudocode of all stages in C#:

```
Write("Enter balance: ");
double n_equity = double.Parse(ReadLine());
Write("\nEnter float risk: ");
int floatrisk = int.Parse(ReadLine());
Write("\nEnter fix risk: ");
int fixrisk = int.Parse(ReadLine());
Write("\nEnter pip amount: ");
float pip_amount = float.Parse(ReadLine());
Write("\nEnter pip amount risk [Low risk>>10 Medium
risk>>100 High risk>>1000] ");
float pip_amount_risk = float.Parse(ReadLine());
pip_amount = pip_amount / pip_amount_risk;
Float temp =(float)(((n_equity * floatrisk) / 100) * fixrisk /
pip_amount);
for (int i = 0; i < fixrisk; i++)
{
temp = temp / 2;
}
WriteLine(temp.ToString("f2"));
```

Result:

The following Figure presents some sample transactions with lot sizes calculated using this formula (algorithm) in Meta Trader 4.

Details:



Gross Profit:	146 293.39	Gross Loss:	14 551.11	Total Net Profit:	131 742.28
Profit Factor:	10.05	Expected Payoff:	484.35		
Absolute Drawdown:	298.28	Maximal Drawdown:	6 314.37 (4.74%)	Relative Drawdown:	4.74% (6 314.37)
Total Trades:	272	Short Positions (won %):	169 (81.66%)	Long Positions (won %):	103 (89.32%)
		Profit Trades (% of total):	230 (84.56%)	Loss trades (% of total):	42 (15.44%)
Largest		profit trade:	5 571.26	loss trade:	-1 853.69
Average		profit trade:	636.06	loss trade:	-346.46
Maximum		consecutive wins ($):	61 (31 209.71)	consecutive losses ($):	10 (-6 314.37)
Maximal		consecutive profit (count):	44 319.31 (31)	consecutive loss (count):	-6 314.37 (10)
Average		consecutive wins:	16	consecutive losses:	3

Figure 1: Meta Trader Maximum DD Performance

As visible, this formula (algorithm) has calculated the maximum drawdown as 4.74%, which is lower than 5%.

Conclusion:

The current formula (algorithm) is more optimized than other common formulas considering this study's sample transaction lot size calculations. The complexity of this formula might be a disadvantage for humans, but it provides higher capital risk management for Martingale or even non-Martingale methods. This formula is better equipped to handle larger capitals; therefore, we cannot guarantee its appropriateness for micro-capitalists.

Funding conflict of interest:

Open all

Ackwonledge:

I intend to say thank's from these persons that supported me as financial for finding and studying mathmatics:

- Ramin Ataiee Zadeh
- Nasrin Fathi
- Afsaneh Ataiee Zadeh
- Farzaneh Ataiee Zadeh
- Iraj Ostovari
- Mr. Mike Andrews (manager at Traders Temple LTD)

Appreciations:

I, Touraj Ostovari, thank the following people for encouraging and assisting me in this great discovery:

- Farzaneh Ataiee Zadeh
- Afsaneh Ataiee Zadeh
- Mr. Mike Andrews

References

1. Systemic Risk in Financial Networks. *Journal of Financial Management, Markets, and Institutions.* I, 129-154 (2013).
2. 1. K. C. T. Duho, J. M. Onumah, R. A. Owodo, E. T. Asare, R. M. Onumah, Bank risk, profit, efficiency and profitability in a frontier market. *Journal of Economic and Administrative Sciences.* **36**, 381-402 (2020).
3. Karim Atashgar, Practical Engineering Statistics, and Probability, Book One, First Edition, Iran University of Science and Technology Publications, Tehran, 2011.
4. Karim Atashgar, Practical Engineering Statistics, and Probability, Book Two, First Edition, Iran University of Science and Technology Publications, Tehran, 2011.
5. Farzin Haji Jamshidi, Engineering Mathematics, First Edition, Avidnegar Publication, Tehran, 2009.
6. Hadi Rangbaran, Mathematics and Its Usage in Economy, Twenty Fourth Edition, Esbat Publication, Tehran, 1967.
7. Mohammad Ghasm Vahidi Asl and Ali Amidi, Mathematical Statistics and Its Applications, Seventh Edition, University Publishing Center Publication, Tehran, 2015.
8. Dr. Rahman Farnosh and Hossin Forotan, Engineering Statistics and Probability, Third Edition, Iran University of Science and Technology Publications, 2014.

Researcher:

Mr. Touraj Ostovari
Date of Birth: 26 October 1996