Military Software Black-Scholes Pricing Model: Value of Software Option and Volatility

Wu Qin, Zhang Yadi

***Abstract*—The idea of real option is applied in military software pricing in this paper. By analysis of Black-Scholes pricing model, we figure two crucial variables. One is value of software option that is regarded as estimated cost in our study. We give the logical analysis based on the option feature of estimated cost and also the practical test by building up a system of equations. The other one is the volatility whose value can be estimated with the method of Monte Carlo analogy. Finally, the application of this military software pricing model is demonstrated in a real case.**

Keywords- military software pricing; Black-Scholes model; price; estimated cost; value of software option; volatility; Monte Carlo analogy

I. INTRODUCTION

As a representative product of the information age, military software has attracted more attention, whether its development process or its cost estimation. And among these aspects of software, the research of cost estimation has shaped well-developed system. For example, the most popular software cost estimation model is COCOMO Ⅱ (Constructive Cost Model Ⅱ), whose key variable is KSLOC (Thousands of Source Lines of Code). Here are the workload equations.

$$PM = A \times (KSLOC)^{E} \times \prod_{i=1}^{17} EM_i \tag{1}$$

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j \tag{2}$$

PM means person-months, EM means effort multiplier and SF means scale effect. There are 17 EM indexes and 5 SF indexes. All the cost indexes are divided into 6 grades as far as their effects to PM are concerned, which are “very low”, “low”, “nominal”, “high”, “very high”, and “extra high”. The numerical value of each effect has been tested by Barry W. Boehm’s work team. The total cost of military software is the result of multiplication of the workload and unit cost [1].

But the estimated cost is not equal to the factual cost of software, and not regarded as the real price of software. So we want to find the hidden relationship between estimated cost and real price, and apply the relationship to military software pricing. This is the new approach to military software pricing.

In the theoretical researches on pricing, with the most application value is option pricing model. And the classical one is Black-Scholes model.

$$V_t^C = X_t N(d_1) - Ke^{-r(T-t)} N(d_2) \tag{3}$$

where

$$d_1 = \frac{\ln(X_t / K) + (r + \sigma^2 / 2)(T - t)}{\sigma\sqrt{T - t}} \tag{4}$$

$$d_2 = d_1 - \sigma\sqrt{T - t} \tag{5}$$

and V_t^C means the t-time value of a call option on the underlying asset with price X_t, K means the exercise price at the mature time T in the future, r means the risk-free rate of return, σ means volatility of underlying asset and $N(.)$ means the standard Normal distribution function [2].

Using the Black-Scholes model in military software pricing, we should solve two problems.

The first problem is how to redefine the variables of the Black-Scholes model, and explain the relationship between the variables and price of software reasonably. There are some successful applications of the real options, and related researches give the detailed analysis about feasibility and logical relationship. For example, study of enterprises human resources value measurement proposes the method based on real options such as Black-Scholes model, and in this study, variable X_t expresses discounted present value of expected cash flow and K expresses related cost [3]. Learning from this idea, we will analyze factors related to military software pricing and the relationship according to Black-Scholes model. In other words, we help the original variables find the exact position in this special situation that is military software pricing.

The second problem is concrete analysis of some special variables. During the process of redefinition and solution, there are two kinds of variable. One is that we cannot give clear definition directly. Firstly, we will assume its definition and get the result. Secondly, we try to find a solution that can avoid the effect of this factor and can also account the price of software. Thirdly, by comparison and analysis we prove the accuracy of previous hypothesis. The other kind is what has no objective value but has clear definition. So we apply another method to account its value. In this way, we can get the complete definition of Black-Scholes model for military software.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. MODELING

A. Conformation of the military software pricing model

Firstly, we study the option features of military software pricing.

In this study, our standpoint is the military. It means this option pricing method can become one of the objective parameters for military software pricing so that the military has the initiative during the negotiations with software development organizations. And there is one important view that military software is not regarded as a product no longer but as a research and development project. It is because of the characteristics of military software such as its high-tech and high-risk. So the price of military software has larger uncertainty, and the military has the right to postpone the software development in order to solve some uncertainty at the moment. Or other, the monopoly and expected return uncertainty determine that the owners or users can wait for the environment improvement and postpone development [4]. We suppose it as the software option, and apply the option pricing model to calculate the real price of military software. Accordingly, it has similar property with the real options, and what as description can be regarded as call option, future cash flow that is price of military software is the underlying asset, period of development is the mature time, and actual cost is the exercise price.

Secondly, we give the military software pricing model based on Black-Scholes model.

$$V = SN(d_1) - Ce^{-r\tau}N(d_2) \tag{6}$$

where

$$d_1 = \frac{\ln(S/C) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \tag{7}$$

$$d_2 = d_1 - \sigma\sqrt{\tau} \tag{8}$$

Combined with the actual needs of military software pricing, we redefine the variables of this model based on their original meanings.

V --value of software option
S --price of software
C -- cost of software development
r -- risk-free rate of return
σ -- volatility of software price
τ -- period of software development
$N(.)$ -- the standard Normal distribution function

B. Declaration of variables

As shown in the key equation (6), there are five factors that will influence the calculation result of software price directly. In order to solve the model, we should clear their values above all. Among these variables, cost is actual cost that is accumulated by the process of software development, and in practice it can be provided as initial data. And the period is as well. We always take the interest on treasury bills as the risk-free rate in the model. There are two variables left that we should pay more attention on: software option value and volatility.

III. ANALYSIS

A. Analysis of software option value

By the analysis during conformation of the pricing model, we can notice that software option is similar to postponement option which is one kind of real options. And in research of real options pricing, this option is always regarded as embedded option and the value of which is regarded as embedded value of underlying asset. By the practical idea, it makes sense that value of software option is embedded value that can be taken as incomplete price, and we consider the estimated cost of software. Like what is accounted by COCOMO Ⅱ, the estimated cost of software is of two characteristics. Firstly, it is based on SDLC (Software Development Life Cycle). This fits the viewpoint in our study that military software is a research and development project. Secondly, estimated cost includes the estimation of fund, time, human resource and other related aspects. And it is similar with opportunity cost of software development. This cost provides the reason that the military decide to postpone the software project or not. If the actual cost is larger than the estimated cost at some stage, this project may be postponed. If the actual cost is lower than the estimated cost all the stage, this project may be implemented as planned. In this sense, the estimated cost is similar to the real options that decide whether to proceed with real investment.

In this way, the variable V in the pricing model (6) can be supposed as estimated cost of software. And it establishes in theory on the basis of the previous analysis. But we need make it clear in practical term especially in demonstration. The idea is that we build up a system of equations to calculate the result when value of variable V is regarded as unknown, and this is one result. We can also get another result after substituting the estimated cost, which is accounted already, in the pricing model. By comparing the two results, we can prove the accuracy of the hypothesis above.

Because the method is based on option pricing model, we consider another kind of option pricing model that is binomial tree. The military software pricing model based on binomial tree has been proposed in Wu Qin's research [1], and we only quote the result as equation (9).

$$MV_5 = \frac{(1+rt_1)(1+rt_2)(1+rt_3)(1+rt_4)(1+rt_5)}{p_1p_2p_3p_4p_5}MV_0 - \frac{q_5}{p_5}(c_1+c_2+c_3+c_4+c_5)$$
$$-\frac{q_4}{p_4}\frac{(1+rt_5)}{p_5}(c_1+c_2+c_3+c_4) - \frac{q_3}{p_3}\frac{(1+rt_4)(1+rt_5)}{p_4p_5}(c_1+c_2+c_3)$$
$$-\frac{q_2}{p_2}\frac{(1+rt_3)(1+rt_4)(1+rt_5)}{p_3p_4p_5}(c_1+c_2) - \frac{q_1}{p_1}\frac{(1+rt_2)(1+rt_3)(1+rt_4)(1+rt_5)}{p_2p_3p_4p_5}c_1 \tag{9}$$

And MV_5 means price of software, MV_0 means value of software option, r means risk-free rate. Of stage i ($i = 1, 2, \cdots, 5$), time is t_i, cost is c_i, success probability or failure probability is p_i or q_i ($p_i + q_i = 1$). And in (9), only

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

two variables MV_5 and MV_0 are unknown. For consistency, we can get another form of (9).

$$S = mV - n \tag{10}$$

$$m = \frac{(1+rt_1)(1+rt_2)(1+rt_3)(1+rt_4)(1+rt_5)}{p_1 p_2 p_3 p_4 p_5} \tag{11}$$

$$n = \frac{q_5}{p_5}(c_1 + c_2 + c_3 + c_4 + c_5) + \frac{q_4}{p_4}\frac{(1+rt_5)}{p_5}(c_1 + c_2 + c_3 + c_4) + \frac{q_3}{p_3}\frac{(1+rt_4)(1+rt_5)}{p_4 p_5}(c_1 + c_2 + c_3) + \frac{q_2}{p_2}\frac{(1+rt_3)(1+rt_4)(1+rt_5)}{p_3 p_4 p_5}(c_1 + c_2) + \frac{q_1}{p_1}\frac{(1+rt_2)(1+rt_3)(1+rt_4)(1+rt_5)}{p_2 p_3 p_4 p_5}c_1 \tag{12}$$

By equation (6), (7), (8), (10), (11), (12), we can get a system of equations of two unknowns (13).

$$\begin{cases} V = SN(d_1) - Ce^{-r\tau}N(d_2) \\ S = mV - n \\ d_1 = \dfrac{\ln(S/C) + (r+\sigma^2/2)\tau}{\sigma\sqrt{\tau}} \\ d_2 = d_1 - \sigma\sqrt{\tau} \\ m = \dfrac{(1+rt_1)(1+rt_2)(1+rt_3)(1+rt_4)(1+rt_5)}{p_1 p_2 p_3 p_4 p_5} \\ n = \dfrac{q_5}{p_5}(c_1+c_2+c_3+c_4+c_5) + \dfrac{q_4}{p_4}\dfrac{(1+rt_5)}{p_5}(c_1+c_2+c_3+c_4) \\ \quad + \dfrac{q_3}{p_3}\dfrac{(1+rt_4)(1+rt_5)}{p_4 p_5}(c_1+c_2+c_3) + \dfrac{q_2}{p_2}\dfrac{(1+rt_3)(1+rt_4)(1+rt_5)}{p_3 p_4 p_5} \\ \quad (c_1+c_2) + \dfrac{q_1}{p_1}\dfrac{(1+rt_2)(1+rt_3)(1+rt_4)(1+rt_5)}{p_2 p_3 p_4 p_5}c_1 \\ p_i + q_i = 1 (i = 1,2,\cdots,5) \end{cases} \tag{13}$$

To solve this system of equations, we get a simultaneous equation (14) by the two key equations (6) and (10).

$$SN(d_1) - \frac{S}{m} - Ce^{-r\tau}N(d_2) = \frac{n}{m} \tag{14}$$

And we use the method of approximation in Excel table to calculate the price of military software S_1.

On the other hand, we account the estimated cost V by COCOMO Ⅱ and substitute it in (6). So we can calculate another result S_2. If S_2 is approximately equal to S_1, we can get the conclusion that it is reasonable for the value of software option to be assumed as the estimated cost in numerical method.

B. Analysis of Volatility

It is very difficult to get objective value of the volatility in Black-Scholes model.

The volatility of underlying asset plays a pivotal role in the Black-Scholes model. In the option theory, the volatility rises, the potential of project appreciation increases and the potential of project depreciation decreases. So the accuracy of volatility directly influences whether the value of option can truly reflect the potential value of project, and serves for decision. However, the real option has the characteristic of non-transaction, and there is no transaction history for underlying asset. So it is very difficult to accurately estimate the volatility and there is a big subjective influence in the current method.

In order to calculate the volatility accurately, we apply the analogue principle of Monte Carlo to forecast the future cash flow of software project, and solve the probability distribution function of project NPV (Net Present Value). In this way we can derive the volatility σ [5].

1) Feasibility of application:

Monte Carlo simulation method supposes that price of investment portfolio changes to a random process, and can be emulated by computer. So the price may produce several paths, and we can structure the remuneration distribution to estimate its value-at-risk. The most common model to select the stochastic process of price is Geometric Brownian Motion which is the theoretical basis of real options [4].

2) Steps of simulation

a) Determine the factors of cash flow and its probability distribution.

b) Random sampling according to the probability distribution, and forecast the cash flow of each year.

c) Calculate a number of NPV according to the simulative cash flow.

d) Calculate the average net present value $\overline{NPV}$ and the standard deviation S.

e) Calculate the all-stage volatility

$$\sigma^* = \frac{S}{\left|\overline{NPV}\right|} \tag{15}$$

f) Calculate the annual volatility

$$\sigma = \sigma^* / \sqrt{n} \tag{16}$$

And the simulation can be realized by Matlab programming. The program code is provided in demonstration.

IV. Demonstration

A. Introduction

This is a real military software project. In this case, the party undertaking research and development projects is a military software constitution of Hang Zhou, and the underlying asset is a kind of military software [1]. There are five stages during the development period: requirements,

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

design, coding, test and maintenance. And the initial data is shown in Table1.

TABLE I. INITIAL DATA

Factors	Stage($i = 1, 2, \cdots, 5$)				
	I require -ments	*II design*	*III coding*	*IV test*	*V mainten -ance*
t_i (year)	0.25	0.25	1	0.5	1
c_i (RMB)	500,000	800,000	2,200,000	1,000,000	1,000,000

From a military software constitution of Hang Zhou.

We want to provide the information that is the software price accounted by Black-Scholes pricing model for the military at date of delivery. In this situation, the time is date of delivery, that is, the development project has already been implemented as planned and what we need to calculate is the finial price of software.

B. Values of variables

Here, the risk-free risk is 5%.

Firstly, we account the estimated cost. It is the estimation of all the development stages and it can be accounted by COCOMO Ⅱ that is not listed in this paper and we only give the calculation result $V = 7,230,554.0$.

Secondly, we calculate the volatility σ by Monte Carlo analogy.

The computational formula of NPV is

$$NPV = \sum_{t=0}^{n} (MV_t - c_t)(1+r)^{-t} \tag{17}$$

So the factors are *MV* and *c* whose distribution are supposed as normal distribution. By the method of Monte Carlo analogy, we give the Matlab program to calculate the volatility which has been shown in Figure1.

The calculation results have been shown in Figure2.

And the estimated results are $\sigma = 0.2839$ when times of simulation are 100, $\sigma = 0.2761$ when times are 1000, $\sigma = 0.2732$ when times are 4000, $\sigma = 0.2720$ when times are 10000. According to these figures, we can find that with the increasing times of simulation, the expectations level off. In this case we take 0.272 as the value of volatility σ.

C. Calculation

We build up the military software pricing model based on Black-Scholes.

$$\begin{cases} V = SN(d_1) - Ce^{-r\tau}N(d_2) \\ d_1 = \dfrac{\ln(S/C) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \\ d_2 = d_1 - \sigma\sqrt{\tau} \end{cases} \tag{18}$$

The calculation process has been shown in Table 2.

TABLE II. CALCULATION

S	d_1	d_2	V
10,000,000	-3.0646	-3.5357	1272.4
100,000,000	1.8229	1.3518	53,421,494.1
50,000,000	0.3516	-0.1195	10,453,866.1
40,000,000	-0.1220	-0.5931	4,966,342.3
45,000,000	0.1280	-0.3431	7,477,198.7
44,000,000	0.0803	-0.3908	6,935,690.7
44,500,000	0.1043	-0.3668	7,204,079.0
44,540,000	0.1062	-0.3649	7,225,755.3
44,548,000	0.1066	-0.3645	7,230,094.2
44,548,800	0.1066	-0.3645	7,230,528.2
44,548,850	0.1066	-0.3645	7,230,555.3
44,548,847.6	0.1066	-0.3645	7,230,554.0

According to it, we obtain the price of military software is 44,548,874.6RMB. In order to test the rationality of the previous hypothesis, we quote the result in Research of Military Software Pricing Based on Binomial tree Method.

The calculation results are that *m* is 6.9526 and *n* is 5752231.37 [1]. By equation (14), we also obtain another result that the price is 44,559,706.6. Comparing it with the previous result that the price is 44,548,874.6, we can find that there is a little difference and get a conclusion that it is reasonable for the value of software option to be supposed as estimated cost.

V. CONCLUSIONS

We propose a new method for military software pricing based on Black-Scholes model and find the solution to the above two problems.

A. The value of software option in Black-Scholes pricing model can be regarded as estimated cost, and it provides a path from cost of software to price of software.

We have designed the option model for military software pricing and found the new way to calculate the price based on the known cost. In the above analysis, the estimated cost is 7,230,554RMB accounted by COCOMO Ⅱ, and the finial price accounted by Black-Scholes pricing model is 44,548,874.6RMB. There is a big difference and it is inaccurate to regard estimated cost as price of military software in the current pricing methods. And it will underestimate the value of software. We apply the real option and the classical option model to calculate the objective price for software, and in other view, we find the mathematical relationship between cost and price shown as Black-Scholes equation.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. After volatility is estimated by Monte Carlo analogy, the new method for military software pricing has been perfected based on Black-Scholes model. And it provides a new field of application for real option.

In original Black-Scholes model, the volatility remains invariant. But in real options, volatility will make great influence in result and it always changes in specific problems. By analysis we make it feasible to use Monte Carlo method. And we give the Matlab program. In addition, there is a extended application for volatility estimation and application of Monte Carlo analogy. During the simulation, we predict the future cash flow of software project which will also provide another according for decision-making in next researches.

REFERENCES

[1] Wu Qin, Wei Ruxiang, "Research of Military Software Pricing Based on Binomial Tree Method," IEEE in press.

[2] Zhang Jing, "Dynamic Methods for Multivariate Option Pricing," 2008 Doctor Thesis.

[3] Liu Dawei, "Study of Enterprises Human Resources Value Measurement in New Economy," CI: F240, u.d.c: 331.021.822.

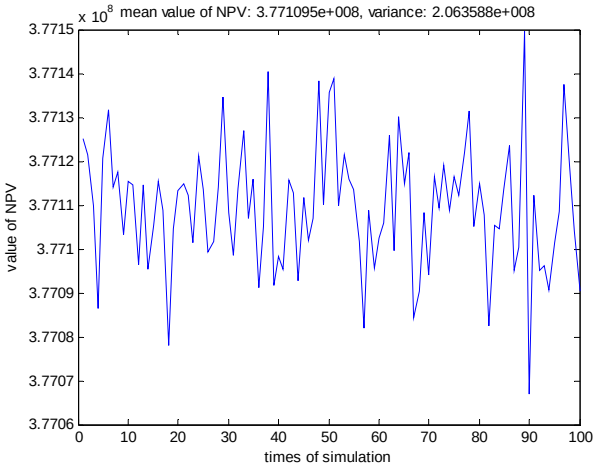
[4] Hu Mingkun, "Evaluation on the Patent Economic Value on the Base of Real Option Theory & Research on the Fluctuation Rate," 2007 Master Thesis.

[5] Ma Zhiwei, Liu Yingzong, "Study on Estimation of Whole Stage Volatility in Real Options," CI: F830.59, pp.1006-7205(2007)02-0028-03.

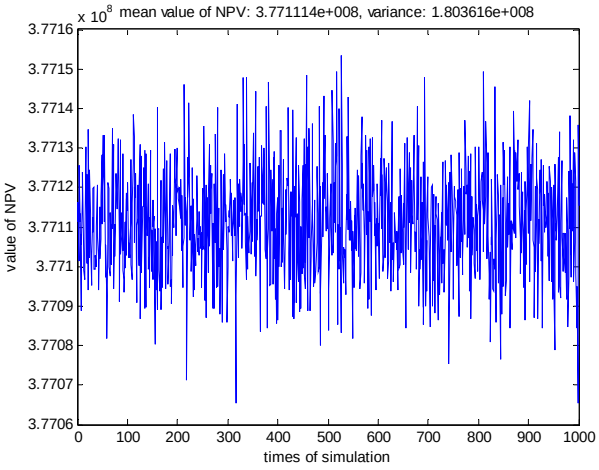
```
clear;
N = 12;     % the length of time in each calculation(unit:3 months)
R = 10000;   % times of repeated simulation

MVdata = [15338958, 28643312, 34403718, 38684162, 44527832];
cdata = [500000, 800000, 2200000, 1000000, 1000000];

meanMV = mean(MVdata);
meanc = mean(cdata);

r = 0.0125;
t = 0:N;
meantNPV = (meanMV - meanc) * sum((1+r).^(-t));
vartNPV = 0.272 * sqrt(N/4) * meantNPV;
varMVc = vartNPV*sum((1+r).^(2*t))/170;
varMV = varMVc/2;
varc = varMVc/2;

MV = random('norm', meanMV, sqrt(varMV), R, N+1);
%figure; plot(reshapt(MV,N*R,1)); xlabel('times of simulation'); ylabel('value of
MV');
c = random('norm', meanc, sqrt(varc), R, N+1);
%figure; plot(reshapt(c,N*R,1)); xlabel('times of simulation'); ylabel('value of
c');
NPV = zeros(1,R);
for i = 1:R
    NPV(i) = sum((MV(i,:)-c(i,:)).*((1+r).^(-t)));
end
meanNPV = mean(NPV);
varNPV = mean((NPV-meanNPV).^2);
varNPV/meanNPV/sqrt(N/4)
figure; plot(NPV); xlabel('times of simulation'); ylabel('value of NPV');
title(sprintf('mean value of NPV: %d, variance: %d', meanNPV, varNPV));
```

Figure 1. Matlab program

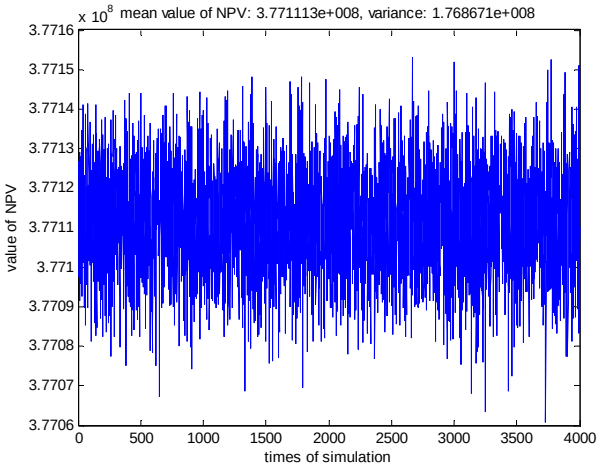
ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



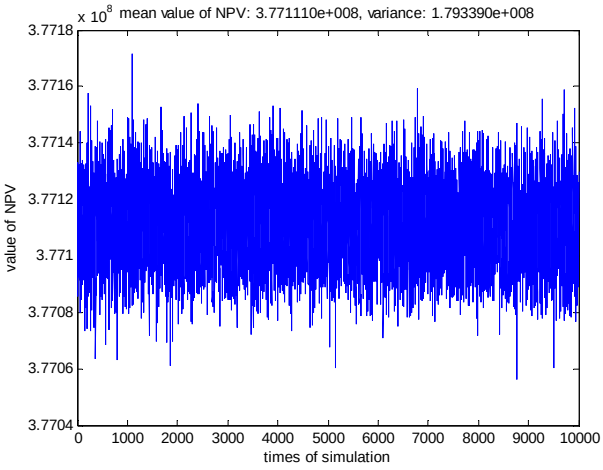
(a)for 100 times



(b)for 1000 times



(a)for 4000 times



(b)for 10000 times

Figure 2. Distribution of NPV value

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)