Optimal Control Methods of Experiment Times in System-of-Systems Combat Computer Simulation

Zhiqiang Zhao[1,a], Feiyue Zhou[2]

[1]*Science and Technology on Complex Land Systems Simulation Laboratory, Beijing, China 100012*
[2]*Southwest Computer Co., Ltd, Chongqing, China 400060*

Abstract. In the process of scheme optimization, in order to eliminate the influence of random factor, it needs to conduct computer simulation of Monte Carlo. Therefore, it is proposed to introduce confidence interval into system-of-systems combat simulation, and confirm whether the Monte Carlo simulation finishes according to data sample generated in simulation process. According to characteristic of data sample, extend correspondingly confidence interval method, and under the condition of obtaining the solution meeting accuracy requirements, reduce simulation experiment times as far as possible. The simulation experiment results show that confidence interval extension method is able to possess self-adaptation control to Monte Carlo simulation.

1 Introduction

The characteristic of weapon system-of-systems combat computer simulation is: Large battlefield scope, long lasting time, wide varieties of weapon, large quantity, complex combat action and various forms of confrontation effects [1]. Due to the limitation of the complexity of systemic confrontation and restrictions of mathematics methods, the ability to establish mathematical models for problems research and to optimize them is very limited, so it needs to introduce emulation technique. In the process of adopting computer simulation technique to research the problem, the simulation model established usually has many random factors; In order to eliminate the influence of random factors to the evaluation of alternative schemes, it needs to rerun the alternative scheme for several times.

Through bibliographic retrieval, the methods used to confirm experiment times n mainly includes 3 kinds [2]: Expert experience method, simple figure line method, confidence interval method. Currently, the expert experience method could be widely used in simulation experiment [3-5]. This method could be simple to use, but not consider the characteristics of model output, therefore, it may cause computing resource waste or inadequate accuracy of solution. In simple figure line method, draw a graph according to accumulative mean value of a series of computational results, and advise to run 10 times at least. The more running times, the flatter the line would be. In this way, the user can select some points which the accumulative mean value tends to be flat in the graph and take it as experiment times. The advantage of such method is easy to understand and execute, just like selecting output interested in decision-making process; the disadvantage is that the user needs to participate in the process, with larger subjectivity.

In confidence interval method, the user is required to give the maximum allowable error in model mean value estimation, which means the accuracy of solution; In process of simulation experiment, adopt running-solving methods for loop iteration until the output result meets solution accuracy. The advantage of such method is to confirm experiment times through statistic inference, and it is a self-adaptive control method; the disadvantage may be that it may lead to the accuracy of the solution is not enough.

In view of the above reasons, the text extends the confidence interval methods to realize the self-adaptive control of experiment times in system simulation optimization.

2 Confidence interval method

2.1 Method introduction

For experimental scheme A, rerun n times, and assume that the value of some performance index which the i time simulation running is corresponding to x_i, then the mean value $x_1, x_2, \cdots, x_n$ of $\overline{x}(n)$ is the unbiased estimation of their expectation. The confidence interval of μ with confidence coefficient of $100(1-\alpha)$ is [6]:

$$\left(\overline{x}(n) \pm t_{n-1,1-\alpha/2}\sqrt{S^2(n)/n}\right) \tag{1}$$

where, n is the time of scheme running; $t_{n-1,1-\alpha/2}$ is

[a] Zhao Zhiqiang: zzqsyp@163.com

distribution of the student with degree of freedom of n - 1, confidence coefficient of 100(1-α); $S(n)$ is the standard deviation of sample x_1, x_2, L, x_n, and the computing method is as follows:

$$S(n)=\sqrt{\sum_{1}^{n}(x_i-\overline{x}(n))^2/(n-1)} \tag{2}$$

From formula (1), the estimated value of relative error:

$$\begin{aligned} r'_{\mu}&=\left|(\overline{x}(n)-\mu)/\overline{x}(n)\le t_{n-1,\alpha/2}\sqrt{S^2(n)}/\overline{x}(n)\right| \\ &=\left|t_{n-1,\alpha/2}\sqrt{S^2(n)}/\overline{x}(n)/\sqrt{n}\right| \end{aligned} \tag{3}$$

Where, $t_{n-1,1-\alpha/2}$ gradually increases little by little with gradual increase of experiment times n, and $\left|\sqrt{S^2(n)}/\overline{x}(n)\right|$ gradually restrains to some value with gradual increase of experiment times n, and r'_{μ} gradually increases less with gradual increase of experiment times n. It can be known from above analysis that in the process of simulation experiment, the confidence interval can be introduced to determine the repetition times n of experimental scheme.

2.2 Finish criteria

In the process of simulation running, the sample data may restrain to some incorrect value earlier, then start to bifurcate. In order to avoid such circumstance, the algorithm needs to set a testing method. When the relative error d_n is less than or equal to accuracy requirement d_r set, adopt such method to test to determine whether it restrains to incorrect value earlier. Specific operations are shown as follows: In case of $r'_{\mu}(n)\le d_r$, continue running experimental scheme n_{oc} times, and if $r'_{\mu}(n+1)$, $r'_{\mu}(n+2)$, ……, $r'_{\mu}(n+n_c)$, re less than or equal to d_r, it deems that the simulation result has reached to the accuracy set and the simulation finishes; Otherwise, in literature[2], define n_{oc} as follows:

$$n_{oc}=\begin{cases} n_{rc} & n\le n_{ref} \\ \mathrm{mod}(n_{rc}n/n_{ref}) & n>n_{ref} \end{cases} \tag{4}$$

Where, n is the running times of current experimental scheme, n_{ref} s the reference times of experimental scheme running, with value setting as 200, and n_{rc} is the testing times n_{ref} corresponding to with value setting as 10.

2.3 Algorithm flow

Step 1. Initialize minimum times $n_{\min}$, reference times n_{ref}, testing times n_{rc} and solution accuracy d_r.

Step 2. Set $n=n_{\min}$, rerun the experimental scheme n times.

Step 3. Through running-solving, carry out loop iteration.

Step 3.1 Calculate mean value $\overline{x}(n)$, with accuracy of $r'_{\mu}(n)$.

Step 3.2 If $r'_{\mu}(n)\le d_{rc}$, transfer to Step 3.4.

Step 3.3 Run the experimental scheme one time, set $n=n+1$, and transfer to Step 3.1.

Step 3.4 Calculate testing times n_{oc}.

Step 3.5 Initialization $i=0$.

Step 3.5.1 Run experimental scheme one time, and set $i=i+1$.

Step 3.5.2 Calculate mean value $\overline{x}(n+1)$, with accuracy of $r'_{\mu}(n+i)$.

Step 3.5.3 In case of $r'_{\mu}(n+i)>d_r$, set $n=n+i$, and transfer to Step 3.3.

Step 3.5.4 In case of $i=n_{oc}$, set $n=n+n_{oc}$, and transfer to Step 4; Otherwise, transfer to Step 3.5.1.

Step 4. Output experimental result $\overline{x}(n)$; Operating n times.

2.4 Experiment and the analysis of results

In system-of-systems combat simulation, the data simple generated in multiple running of single scheme are mainly subject to two kinds of random distribution: Normal distribution, Bernoulli distribution. Therefore, in order to test the performance of algorithm, use data sample generated in static state ($N(\mu, \sigma)$ means normal distribution; $B(1, \mu)$ means Bernoulli distribution) listed in Table 1 to conduct experiment; Set the confidence coefficient of sample mean expectation μ as 0.9, that is set solution accuracy as 0.05; n_{min} is set as 5. In order to eliminate the influence of data series, in experiment, each static distribution randomly generates 1000 groups of different data series.

Table 1 Test result of artificial models to algorithm performance

No	Static Dis	n_A	P_A	n_E
1	$N(1,0.2)$	55.424	0.929	47.00
2	$N(1,0.4)$	186.927	0.906	173.19
3	$N(1,0.6)$	411.896	0.908	389.67
4	$N(1,0.8)$	738.513	0.905	692.74
5	$N(1,1)$	1155.819	0.906	1082.41
6	$B(1,0.9)$	108.243	0.716	120.15
7	$B(1,0.7)$	480.194	0.903	464.35
8	$B(1,0.5)$	1137.156	0.904	1082.41
9	$B(1,0.3)$	2640.535	0.890	2525.26
10	$B(1,0.1)$	8173. 41	0.717	9741.69

In the process of algorithm performance testing, the data sample size required may be much larger than 46, but currently available data only gives the data of t distribution in $n < 46$; Therefore, when $n > 46$, the algorithm introduces $t_\alpha(n)$ approximate calculation of solving way given in literature[7], namely $t_\alpha(n) = z_\alpha$. Though there may be certain error for adopting such approximate value, its relative error shall not exceed 1.3% at most; Therefore, the influence to experiment result is very little. In Table 1, n_A means the mean value of experiment times obtained by algorithm; P_A is the probability meeting $\left|(x(n)-\mu)/\mu\right| < d_r$ in the mean value obtained; n_E means the experiment times to be conducted theoretically, namely:

$$n_E = t^2_{n-1,1-\alpha/2}\sigma^2 / (\mu^2 d_r^2)$$

According to Table 1, when the data sample generated in Monte Carlo simulation process is subject to normal distribution, the confidence interval method is adopted and all solutions can reach to accuracy requirements; But if the data sample generated in Monte Carlo simulation process is subject to Bernoulli distribution, the confidence interval method is adopted, and partial solutions fail to reach to accuracy requirements, which mainly because the value of data sample only has 0 and 1. If the uniformity of data sample generated in the initial phase of Monte Carlo simulation is poor, it is easy to cause earlier finish of simulation. For example, data sample subjects to *B*(1, 0.9), but the first 15 samples generated by simulation are {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}, it is easy to obtain that the sample mean value as 1 according to algorithm.

3 Confidence interval extension method

3.1 Algorithm extension

3.1.1 Normal distribution

Through analysis, we could see that: In the process of simulation experiment, if the data sample generated possesses good uniformity, the sample mean value generated by a small amount of data samples should meet solution accuracy. Based on this, the uniformity of data sample is deemed as another criterion to determine whether algorithm for solving has finished in this text.

For normal distribution, there are 3 special values, that is, sample mean value $\overline{x}$, sample mean value x_M , interval mean value x_A. For the data sample, according to the sequence of from small to large, the computational algorithm of sample mean value x_M is as follows:

$$x_M = \begin{cases} (x_{n/2} + x_{n/2+1})/2 & n\%2 = 0 \\ x_{(n+1)/2} & else \end{cases} \quad (5)$$

The computational algorithm of interval mean value x_A is as follows:

$$\begin{cases} x_A = x_l + x_h \\ l = \mathrm{mod}(n(1-\Phi(2))+1) \\ h = \mathrm{mod}(n\Phi(2)-1) \end{cases} \quad (6)$$

where, n is the number of data sample.

$$\Phi(2) = \int_{-\infty}^{2} \frac{1}{\sqrt{2\pi}} e^{-\mu^2/2} d\mu = 0.9772$$

If the data sample generated by simulation experiment possesses good uniformity, the above three values are approximately equal. Based on this, adopt $\overline{x}$, x_M , x_A to test the uniformity of data sample. The specific method is to execute following operations after each simulation running:

- Calculate the maximum value $x_{\max}$ and minimum value $x_{\min}$ in solution $\overline{x}$, x_M , x_A after each simulation running.
- When $\left|(x_{\max} - x_{\min})/x_{\min}\right| < d_r/2$, register $n_{node} = n_{node} + 1$; Otherwise, register $n_{node} = 0$.
- When it comes to $n_{node} = n_{oc}$, the data sample has good uniformity, and the simulation running is finished; Otherwise, continue simulation running.

3.1.2 Bernoulli distribution

For any data sample subject to a certain random distribution, the larger the sample size is, the better overall uniformity of data sample would be, namely, the mean value and expectation of data sample more close to theoretical value. Based on this, in this text, increase the value of minimum times to solve the deficiency of partial solution not meeting accuracy requirements due to poor uniformity of data sample generated at the initial phase of Monte Carlo simulation in this text. When the value of $n_{\min}$ is set as 0.9, meet the experiment times theoretically required by accuracy requirements, specific computing method as follows:

$$\begin{aligned} n_{\min} &= t^2_{n-1,1-\alpha/2}\sigma^2 / (\mu^2 d_r^2) \\ &= t^2_{n-1,1-\alpha/2}(0.9\times 0.1)/(0.9^2 d_r^2) \approx z^2_{\alpha/2}/(9d_r^2) \end{aligned} \quad (7)$$

3.2 Algorithm flow

Step 1. Initialize minimum times $n_{\min}$, reference times n_{ref} , testing times n_{rc} and solution accuracy d_r .

Step 2. Set $n = n_{\min}$, rerun the experimental scheme for n times.

Step 3. if the data sample is subject to normal

distribution, transfer to Step5.

Step 4. Adjust $n_{\min}$, rerun experiment scheme $n_{\min} - n$, and set $n = n_{\min}$.

Step 5. Through running-solving, carry out loop iteration.

Step 5.1 Calculate mean value $\overline{x}(n)$, with accuracy of $r'_{\mu}(n)$.

Step 5.2 If the data sample is subject to Bernoulli distribution, transfer to Step5.4; Otherwise, adopt the method in 3.1.1 to test the data sample, when meet requirements, register $n_{node} = n_{node} + 1$; Otherwise, register $n_{node} = 0$.

Step 5.3 Calculate testing times n'_{oc}, in case of $n_{node} = n'_{oc}$, transfer to Setp6.

Step 5.4 In case of $r'_{\mu}(n) \le d_r$, transfer to Step 5.6.

Step 5.5 Run the experimental scheme one time, set $n = n + 1$, and transfer to Step 5.1.

Step 5.6 Calculate testing times n_{oc}.

Step 5.7 Initialize i=0.

Step 5.7.1 Run experimental scheme one time, and set $i = i + 1$.

Step 5.7.2 Calculate mean value $\overline{x}(n+i)$, with accuracy of $r'_{\mu}(n+i)$.

Step 5.7.3 If the data sample is subject to Bernoulli distribution, transfer to Step5.7.5; Otherwise, adopt the method in 3.1.1 to test the data sample, when meet requirements, register $n_{node} = n_{node} + 1$; Otherwise, register $n_{node} = 0$.

Step 5.7.4 Calculate testing times n'_{oc}, in case of $n_{node} = n'_{oc}$, transfer to Setp6.

Step 5.7.5 In case of $r'_{\mu}(n+i) \le d_r$, set $n = n + i$ and transfer to Step5.5.

Step 5.7.6 In case of $i = n_{oc}$, set $n = n + n_{oc}$, and transfer to Step 6; Otherwise, transfer to 5.7.1.

Step 6. Output experimental result $\overline{x}(n)$; Operating n times.

3.3 Simulation experiment

In order to test the validity of algorithm, conduct experiment to static distribution listed in Table 1 again, and set the solution accuracy as 0.05 and 0.10. The results of the experiment are shown in Table 2.

It can be known through comparison of data in Table 1 and Table 2 that, after algorithm extension, when the data sample generated in the Monte Carlo simulation process is subject to normal distribution, under the circumstance of ensuring the accuracy requirements of solution, the times of simulation experiment can be reduced; When the data sample generated in the Monte Carlo simulation process is subject to Bernoulli distribution, it can reach to accuracy requirements of solution, and when the mean value is less than 0.9, the difference between simulation experiment times and theoretical value is small.

Table 2 Test result for artificial models to the extension of algorithm performance

No	Static Dis	$n_A = 0.05$	$P_A = 0.05$	$n_A = 0.10$	$P_A = 0.10$
1	$N(1, 0.2)$	51.960	0.925	18.516	0.960
2	$N(1, 0.4)$	151.059	0.907	39.881	0.907
3	$N(1, 0.6)$	296.566	0.908	94.988	0.924
4	$N(1, 0.8)$	508.675	0.911	152.452	0.912
5	$N(1, 1)$	751.777	0.909	220.263	0.916
6	$B(1, 0.9)$	145.274	0.937	36.051	0.944
7	$B(1, 0.7)$	488.995	0.903	126.074	0.869
8	$B(1, 0.5)$	1138.806	0.905	286.652	0.914
9	$B(1, 0.3)$	2659.756	0.905	667.635	0.901
10	$B(1, 0.1)$	10228	0.906	2437.835	0.852

4 Conclusion

Add the confidence interval extension method into system-of-systems combat simulation as a module of management and control of member, which can self-adaptively control the times of Monte Carlo emulation, so that the solution calculated can meet accuracy requirements and the computing resource will not be wasted. However, since the rerunning times of scheme is directly proportion to $\sigma^2 / (\mu^2 d_r^2)$, it needs to cautiously give corresponding d_r for estimation of σ and μ before system-of-systems combat simulation to avoid the waste of computing resource.

References

1. Z.Q Zhao, K.D Huang, Z.R. Ni. J. Sys. Simu. **20**, 1103, 2008.
2. K Hoad, S Robinson, R Devies. Proc. The Winter Simulations Conference, 505, 2007.
3. S.F Zhang, H Cai. J Sys. Simu. **12**, 54.
4. H Wu, J.L Shaprio, W.A. Sci. Eng. Tech. **21**, 51 , 2005.
5. J April, M Better, F Glover, et al, Proc. The Winter Simulation Conference, 642, 2006.
6. X.M Zhang. <http://rave.ohiolink.edu/etdc/veiw?acc_num=ohiou1184619898>.
7. Z Sheng, S.Q Xie, C.Y Pan. *Probability Theory and Statistic*. HE Press, Beijing China, 1989.