The Application of Genetic Algorithm in the Optimal Offer of Stochastic Number of Bidders

Guodong Chen
School of Management and Economics
Beijing Institute of Technology
BeiJing, China
Chenguodong1979@126.com

***Abstract*—Competitive bidding is widely used in international equipment procurements and construction projects for the moment. With the development of market economy in our country, competitive bidding is introduced in various industries, especially popular in the large-scale construction projects and government procurements. Thus the bidding theory becomes the focus of enterprises' decision-making. In this paper, the competitive bidding with a stochastic number of bidders is considered. Under the symmetric situation of bid the paper gets a quotation equation and then provides a new solution based on genetic algorithm.**

Keywords- competitive bidding; optimal offer;

I. INTRODUCTION

Competitive bidding is widely used in international equipment procurements and construction projects for the moment. With the development of market economy in our country, competitive bidding is introduced in various industries. Especially in recent years, bidding is widely used in the large-scale construction projects and government procurements at home. The owners mostly adopt the competitive bidding, forcing the bidders to get contracts through a fierce competition, so how to make bidding strategies becomes the focus of each enterprise.

Competitive bidding refers that the tenderer invites several or even dozens of bidders to participate in the bidding and selects the best bidder for the tenderer to make a deal through the competition of many bidders. The international competitive bidding has two ways: ① the open tender: The open tender is a kind of infinite competitive tender. The tenderer needs to publish tender advertisements on the major media at home and abroad when they adopt this way. All of the people who are interested in the tender have an opportunity to buy the tender material to take part in the bidding. ② the selective tender: It ' s a kind of limited competitive tender. The tenderer invites businessmen according to their specific business relationship and information instead of advertising for it in newspapers. After the prequalification, the businessmen who are selected can take part in the bidding.

At present, most of the papers discuses the case with a known fixed number of bidders, in fact the number of potential competitors is often uncertain. This paper studies the competitive bidding with stochastic number of bidders and provides an optimal decision-making strategy for bidding under this circumstance and finally provides a new solution based on genetic algorithm[1-3].

II. THE COMPETITIVE BIDDING WITH STOCHASTIC NUMBER OF BIDDERS

A. Non-Symmetric Situation

Assuming that there are numbers of m +1 potential competitor in a bidding. Select a bidder, which will be called a reference bidder later, to analyze, and the rest are called competing bidders. m is a random variable. On the basis of the historical data, the reference bidder masters the quotations that can win the biddings of all previous projects biddings, each competitor's quotation and the estimated value of cost on his own side.

Supposing that the reference bidder's estimated value of the project is c_0; quoted price is b_0 and profits is p_0. According to the historical data, work out the ratio Y_i of the quoted price b_i of the i th competitor to the reference bidder's estimated value c_0, $Y_0 = b_0 / c_0$, $Y_i = b_i / c_0$, $i = 1, 2, \cdots, m$. Usually Y_i is a random variable that obeys a certain distribution. The reference bidder takes part in the bidding with the quoted price of b_0, the revenue function is $p_0 = b_0 - c_0$, when $Y_0 < Y_1, Y_2, \cdots Y_m$, p_0 =0.

Supposing that f_{Y_i} is a probability density function of the random variable Y_i. Due to the low bid in the competitive bidding, in order to analyze this conveniently, we define the function F_{Y_i} as $F_{Y_i}(Y_0) = \int_{Y_0}^{\infty} f_{Y_i}(Y_i) dY_i$, which means the probability of assigning a value that bigger than Y_0 for the random variable Y_i.

Supposing that m is a random variable obeying Poisson distribution and the parameter is λ. The probability density function of the random variable m is $g(m) = \frac{\lambda^m \times e^{-\lambda}}{m!}$,

m =0, 1, 2⋯If the reference bidder takes part in the bidding with quoted price b_0, its predicted revenue can be expressed by

$$E(p_0)=(b_0-c_0)\times\left[\sum_{m=0}^{\infty}\frac{\lambda^m\times e^{-\lambda}}{m!}\times\prod_{i=0}^{m}F_{Y_i}(Y_0)\right] \quad (1)$$

In the formula, $\prod_{i=0}^{m}F_{Y_i}(Y_0)$ means the probability that the quoted price of other bidders is higher than that of the reference bidder b_0.

Assuming that the estimated cost c_0 is a constant for a given project and then we can work out b_0, which make $E(p_0)$ at the maximum point. Namely calculate

$$\underset{b_0\geq c_0}{E(p_0)}=(b_0-c_0)\times\left[\sum_{m=0}^{\infty}\frac{\lambda^m\times e^{-\lambda}}{m!}\times\prod_{i=0}^{m}F_{Y_i}(Y_0)\right] \quad (2)$$

Notice that for all $b_0<c_0$, there always exists $E(p_0)$ <0, so we can work out the maximum value according to the unconstrained problems.

Theoretically, from the first order necessary condition, make $\frac{dE(p_0)}{db_0}=0$, we can work out the optimal offer b_0. In practical applications, it is nearly impossible to work out the analytical solution. As for such complex optimization problems, it is ideal to make use of the genetic algorithm. The following we will illustrate this point with symmetric situation.

B. The symmetric situation

Below we consider a simplest situation, all functions F_{Y_i} are same, i.e., the symmetric situations. Assuming that $F_{Y_i}=F(Y)$, then the expected revenue of reference bidder is: $E(p_0)=(b_0-c_0)\times\left[\sum_{m=0}^{\infty}\frac{\lambda^m\times e^{-\lambda}}{m!}\times\prod_{i=0}^{m}F_{Y_i}(Y_0)\right]$.

From the assumption we can know $F_{Y_i}=F(Y)$, so

$$E(p_0)=(b_0-c_0)\times\left[\sum_{m=0}^{\infty}\frac{\lambda^m\times e^{-\lambda}}{m!}\times F^m(Y_0)\right]$$

$$=(b_0-c_0)\times e^{-\lambda}\times\sum_{m=0}^{\infty}\frac{(\lambda F(Y_0))^m}{m!}$$

$$=(b_0-c_0)e^{-\lambda}\times e^{\lambda F(Y_0)}$$

$$=(b_0-c_0)\times e^{-\lambda(1-F(Y_0))} \quad (3)$$

Take the derivative about b_0 of both sides of the above formula, we can get

$$\frac{dE(p_0)}{db_0}=e^{-\lambda(1-F(Y_0))}\left[1-\left(\frac{b_0}{c_0}-1\right)\times\lambda f(Y_0)\right]$$

Make $\frac{dE(p_0)}{db_0}=0$, i.e.,

$$e^{-\lambda(1-F(Y_0))}\left[1-\left(\frac{b_0}{c_0}-1\right)\times\lambda f(Y_0)\right]=0 \quad (4)$$

Because $e^{-\lambda(1-F(Y_0))}$ is bigger than 0, so

$$1-\left(\frac{b_0}{c_0}-1\right)\times\lambda f(Y_0)=1-(Y_0-1)\times\lambda f(Y_0)=0 \quad (5)$$

Hence, the optimal offer we get is the product of Y_0 that satisfies the formulas from 1 to 5 and the estimated cost. This offer can make the predicted value reach the maximum point.

III. THE INTRODUCTION OF THE GENETIC ALGORITHM

Genetic algorithm (GA) is a random search method based on the biological laws of evolution. The professor of United States J.Holland came up with it first in 1975. Operating on the object structure directly is its main characteristic. It can gain and guide the optimal searching space automatically and adjust the searching direction adaptively. There doesn't exist limitation of derivation and continuity of function. These properties of the genetic algorithm have been widely used in combinatorial optimization, machine learning, signal processing, adaptive control, industrial engineering and other fields.

The genetic algorithm is similar to biological evolution, looking for good chromosomes to deal with problems by acting on the genes of the chromosomes. It needs to do nothing but evaluate every chromosome produced by the algorithm and select good chromosomes based on the adaptability to make them have more chances to reproduce finally. The genetic algorithm produces digital codes of several problems to be solved in a random fashion. These digital codes are chromosomes and they will form an initial population. Then they give a numerical evaluation to each individual via fitness function, selecting the individuals of high fitness to take part in the genetic operations. After that, the individuals will generate a new population. Finally let this new population be in next evaluation. That is the basic theory of the genetic algorithm.

A genetic algorithm is constructed by stochastic operators, and its robust search ability is based on the theorem depicted in, which states, "short schemata of low order with aptitude above average, and exponentially increase its number by generations", this is:

$$m(H,t+1)\geq(H,t)\frac{f(H)}{f_{avg}}\left[1-p_c\frac{\delta(H)}{l-1}\right]-O(H)p_m \quad (6)$$

where $m(H,t+1)$ and $m(H,t)$ are the schemata number H in the generation $(t+1)$ and t respectively, $f(H)$ is the average aptitude value of the strings that is included on the schemata H, f_{avg} is the total population's average aptitude value, l is the total string length, $\delta(H)$ is the schemata length from H, $O(H)$ is the schemata order from H, p_c is the crossover probability and p_m is the mutation probability.

A. A genotypic viewpoint of genetic algorithm

The traditional biological viewpoint is to consider the contents of the population as a gene pool and study the transitory and equilibrium properties of gene value (allele) proportions over time. In the case of a canonical GA, it is fairly easy to get an intuitive feeling for such dynamics. If fitness-proportional selection is the only active mechanism, an initial randomly generated population of N individuals fairly rapidly evolves to a population containing only N duplicate copies of the best individual in the initial population. Genetic operators like crossover and mutation counterbalance this selective pressure toward uniformity by providing diversity in the form of new alleles and new combinations of alleles. When they are applied at fixed rates, the result is a sequence of populations evolving to a point of dynamic equilibrium at a particular diversity level. Figure 2 illustrates how one can observe these effects by monitoring the contents of the population over time.

Note that, for the fitness landscape described above, after a relatively few numbers of generations almost all individuals are of the form 1, after several hundred generations as we see that, although more than 50% of the gene pool has converged, the selective pressures on the remaining alleles are not sufficient to overcome the continual diversity introduced by mutation and crossover, resulting in a state of dynamic equilibrium at a moderate level of genetic diversity[4-7].

Gen 0	**Gen 5**	**Gen 50**	**Gen 200**	**Gen 500**
01111100010011	11000001011100	11111101010101	11111111010011	11111111011110
00010101011100	11100001000010	11111110011101	11110111011110	11111111110011
01001000110010	11110100010100	11111011001001	11111011010110	11111110010001
00110001110010	11110111100011	11111101000001	11111110101011	11111111011010
...	...	...	...	...
00010001111110	11110111100001	11111111010011	11111111011001	11111111110000
10101111100111	11110111011100	11111111010011	11101111111001	11111111010001
01111101010100	11101010100100	11110100000111	11111111100011	11111111011000
11000011100110	11010100011000	11111101011001	11111111110011	11111111010001
11001101100001	11100001110001	11111100101001	11111111111011	11111111110110
11110111011100	11110110011011	11111011100101	11111001001010	11111111110010
11001010100000	11010101000001	11111100000001	11111110111010	11111111110100
00101000101111	11010001010110	11111111100101	11111110011011	11111111110011
01000011111110	11110100010101	11111111001001	11111110110111	11111111111000

Figure 1. A genotypic view of a GA population.

B. An optimization viewpoint

There are of course other ways of monitoring the behavior of a GA. We might, for example, compute and plot the average fitness of the current population over time, or the average fitness of all of the individuals produced. Alternatively, we might plot the best individual in the current population or the best individual seen so far regardless of whether it is in the current population. In general such graphs look like the ones in Figure 2 and exhibit the properties of "optimization graphs" and "learning curves" seen in many other contexts.

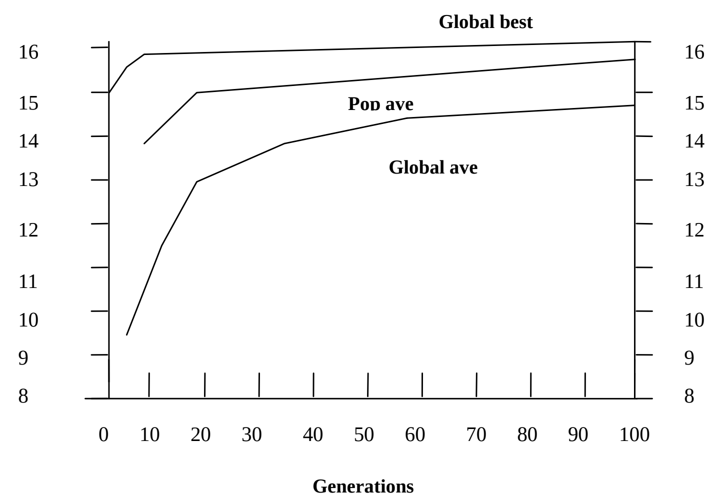

Figure 2. A genotypic view of a GA population.

Measurements of these sorts have encouraged the view of GA as function optimizers. However, one must be careful not to adopt too simplistic a view of their behavior. The genotypic and phenotypic viewpoints provide ample evidence that populations do not in general converge to dynamic steady states containing multiple copies of the global optimum. In fact it is quite possible for the best individual to never appear or appear and then disappear from the population forever. With finite and relatively small populations, key alleles can be lost along the way due to sampling errors and other random effects such as genetic drift. In theory, as long as mutation is active, one can prove that every point in the space has a non-zero probability of being generated. However, if the population evolves toward subspaces not containing the optimum, the likelihood of generating optimal points via mutations of the current population becomes so small (e.g., < 10−10) that such an event is unlikely to be observed in practice.

With a canonical GA every point generated (good or bad) has a relatively short life span since the population is constantly being replaced by new generations of points. Hence, even if a global optimum is generated, it can easily disappear in a generation or two for long periods of time.

These observations suggest that a function optimization interpretation of these canonical GA is artificial at best and raise the question as to whether there might be a more natural behavioral interpretation.

C. *The basic operation processes of genetic algorithm*

The basic operation processes of genetic algorithm are just as follows[8-14]:

1) *Initialization:*Setting up the evolution algebra counter for t =0 and setting up the maximum evolution algebra for T , then regarding the randomly generated individual M as the initial population $P(0)$.
2) *Individual Evaluation:*Calculating the fitness of each individual in the population $P(t)$.
3) *Selecting Operation:*Let the selection operator act upon the population. The purpose of the choice is bringing the good individual into the next generation directly or bringing the new individual that produced by matching and crossing into the next generation. The selecting operation is based on the evaluation of the fitness of individuals in the population.
4) *Cross Operation:*Let the cross operator act upon the population. The cross refers to the operation that replacing and restructuring the partial structures of two original individuals and then generating new individuals. The cross operator plays a core role in the genetic algorithm.
5) *Variating Operation:* Let the mutation operator act upon the population. That is to say, change the genic value of some loci of individual string in population. After selection, crossover and mutation operation, the population $P(t)$ gets the next generation population $P(t+1)$.
6) *Termination of the Condition Judgment:*If we can regard the individual that equipped with the maximum fitness as the most optimal solution and then output it. Terminate calculation finally.
7) Or the genetic algorithm procedures can be simplified as follows:
   a) Choose the initial population of individuals.
   b) Evaluate the fitness of each individual in that population.
   c) Repeat on this generation until termination (time limit, sufficient fitness achieved, etc.):

ⅰ. Select the best-fit individuals for reproduction.

ⅱ .Breed new individuals through crossover and mutation operations to give birth to offspring.

ⅲ. Evaluate the individual fitness of new individuals.

ⅳ. Replace least-fit population with new individuals.

The flow chart of Genetic Algorithm is shown as figure 3.

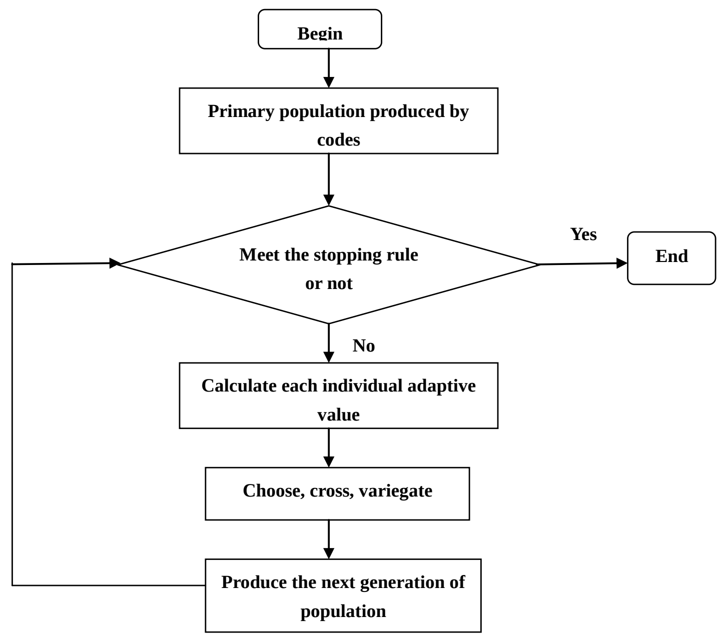

Figure 3. The flow chart of genetic algorithm.

IV. THE EXAMPLE OF BIDDING DECISION-MAKING BASED ON STOCHASTIC NUMBER OF BIDDERS

An enterprise takes part in a bidding war. The estimated value of cost on the project c_0 is 10 million yuan. The number of potential competitors m is a random variable that obeys Poisson distribution.

According to historical data, this enterprise knows m obeys the Poisson distribution, whose parameter λ = 15. And according to the previous situation of bidding on the similar projects, the enterprise thinks that the ratio of each competitor's quoted price and it's own estimated cost obeys the normal distribution, whose mean is 1.45 and standard deviation is 0.3. The aim of the tenderer is to select the quoted price, which can make his expected revenue be in maximum point.

$$g(m)=\frac{15^m \times e^{-15}}{m!},\ m=0,1,2,\cdots \tag{7}$$

$$f(Y)=\frac{1}{\sqrt{2\pi}\times 0.3}e^{\left(-\frac{(Y-1.45)^2}{2\times 0.3^2}\right)} \tag{8}$$

Bring $g(m)$ and $f(Y)$ into the equations（1-5）, we can get:

$$1-\left(\frac{b_0}{c_0}-1\right)\times\lambda f(Y_0)=1-\left(Y_0-1\right)\times\lambda f(Y_0)=0$$

$$1-\left(Y_0-1\right)\times 15\times\frac{1}{\sqrt{2\pi}\times 0.3}\exp^{\left(-\frac{(Y_0-1.45)^2}{2\times 0.3^2}\right)}=0 \tag{9}$$

The analytic solution of this equation is difficult to work out. This paper deals with the problem by means of genetic algorithm and programs it in Matlab.

First, build a bid.m file to describe this problem, and the codes are as below:

```
function r=bid(y)
   r=1-(y-1)*15*(1/(sqrt(2*pi)*0.3))*exp(-(y-
1.45)^2/(2*0.3^2));
   r=abs(r);
```

Then establish a gabid.m file to run the genetic algorithm, codes are as follows:

```
nvars = 1;
options = gaoptimset;
%% change the parameter
options = gaoptimset(options,'Display', 'off');
[x,fval,exitflag,output,population,score] = ...
ga(@bid, nvars,[],[],[],[],1,2,[],options);
```

After running, $Y_0 =$ 1.0992, so the optimal quoted price of this enterprise is 10,992,000 yuan.

V. CONCLUSION

With the development of market economy in our country, the competitive bidding is introduced in the lager-scale construction projects and government procurements. Therefore, the bidders pay much more attention to how to make bidding strategy to get the maximum expected revenue. As a result, the bidding theory is becoming a hot topic of economic scholars day by day. This paper studies the competitive bidding problems with stochastic number of bidders, introducing the genetic algorithm and programming in Matlab, and achieves the optimal strategy of quotation.

REFERENCES

[1] Hanmin Liu, Qinghua Wu, and Xuesong Yan, "Solve Combinatorial Optimization Problem Using Improved Genetic Algorithm", AISS: Advances in Information Sciences and Service Sciences, Vol. 5, No. 1, pp. 1 ~ 8, 2013

[2] YANG Zhen, WANG Cheng-Jun, and GUO Li, "Optimization for the Row Heights in Medium-Length Hole Blasting Design by Genetic Algorithms", AISS: Advances in Information Sciences and Service Sciences, Vol. 5, No. 1, pp. 9 ~ 15, 2013

[3] Daniela di Serafino, Susana Gomez, Leopoldo Milano, Filippo Riccio, and Gerardo Toraldo, “A genetic algorithm for a global optimization problem arising in the detection of gravitational waves”, Journal of Global Optimization, vol. 48, no. 1, pp.41-45, 2010.

[4] Vose, M. D., The simple genetic algorithms: foundations and theory. The MIT Press, USA, 1999.

[5] Homaifa, A., Lai, S. and Qi, X., “Constrained optimization via genetic algorithms”, Journal of Simulation, vol. 62, no. 4, pp.242-254, 1994.

[6] Z Michalewicz, Genetic Algorithms + Date Structure = Evolution Program, 3th Edition, Springer - Verlag, German, 1996.

[7] J. Kingdon, K. Feldman, “Genetic algorithms and applications to finance”, Journal of Applied Mathematical Finance, vol. 2, no. 2, pp.89-116, 1995.

[8] Glynn T. Tonsor and Terry Kastens, “Do starting values really matter? Development of a genetic algorithm approach”, Journal of Applied Economics Letters, vol. 16, no. 8, pp.781-784, 2009.

[9] Arthur Schwartz, “Ethics in competitive bidding and contracting”, Journal of Science and Engineering Ethics, vol. 10, no. 2, pp.277-282, 2004.

[10] Ralph Berndt, Internationale Wettbewerbsstrategien, Springer Berlin Heidelberg, Germany, 2007.

[11] Emin Erkan Korkmaz, “Multi-objective Genetic Algorithms for grouping problems”, Journal of Applied Intelligence, vol. 33, no. 2, pp.179-192.

[12] Joshua W. Hallam, Olcay Akman, and Füsun Akman, “Genetic algorithms with shrinking population size”, Journal of Computational Statistics, vol. 25, no. 4, pp.691-705.

[13] Koh Sueyi, Leowsoo Kar, and Loke Kar Seng, Artificial Intelligence Applications and Innovations, Springer US, Germany, 2005.

[14] John J. Grefenstette, Analogical and Inductive Inference, Springer Berlin Heidelberg, Germany, 1992.