LDAGENT GROUP32

|  |  |  |
| --- | --- | --- |
|  | LAI WEIZHONG 28969367 [wl4a17@soton.ac.uk](mailto:wl4a17@soton.ac.uk) |  |

1 INTRODUCTION

LDAgent try his best to maximize the benefits of choice, the ultimate goal is to stay as close as possible to the Nash equilibrium. Meanwhile, the agent will have made a choice to maximize the benefits of it’s own. I try to use Genetic Algorithm to select the choice of maximal benefits. After many times of biding, the Genetic Algorithm will select a better genetic(bid). And I also add some a little more randomness to the algorithm, so I add the Gaussian distribution to the genetic algorithm.

**2. GENETIC ALGORITHM**

Genetic algorithm originated from computer simulations of biological system. It is a random global search and optimization method that mimics the biological evolutionary mechanism in nature and draws on Darwin’s theory of evolution and Mendelian’s theory of genetics. Its essence is an efficient search method, which can automatically acquire and accumulate knowledge of search space in the search process and adaptively control the search process to get the best solution.

3 IMPLEMENTATION

3.1 Generate Bid

use a random bid function “generateRandomBid()” to generate a ramdom bid. I will give this function upper and lower limits and if the random generate bid function generate a bid which cater to the utility between the upper utility and lower utility, the function will return that bid. Use this method do not have to care about how to bid, only need to care about the utility. Although this method reduces the difficulty of the code, there are some detail can not handle, for example, the same utility may have different bid, so this method has a lot of randomness.

3.2 Different time have different ways to bid

I divide the time into two equal parts, when time less than 0.5, LDAgnet tentatively bid, but not always bid 1. I want to confuse opponents, let opponents can not guess my way of biding. I also want to know the bottom line of the opponent’s compromise

3.3 Genetic Algorithm

*3.3.1 calculate fitness.* As the utility always be 1 is likely to result in o compromises for all agents. Thus, it will lead to 0 utility and 0 far from Nash Equilibrium. So I decided to make the value of adaptability gradually decrease over time. And I also make the fitness value is a random value subject to Gaussian distribution and the means of Gaussian distribution is decrease as the time goes. So the values of fitness calculated in this way are both random changes and not always 1. Although there is randomness, the probability of obtaining high benefits after a large umber of bids is high in terms of probability.

*3.3.2 Choose a good gene* Use Roulette selection method to choose. First, calculate the probability that each of the 10 genes will be selected. f(xi) represents the fitness value. The fitness of each gene closer to the calculated fitness that the basic probability of being selected is high. And then, put the probability of all genes in an cumulative array. The nth element of the array represents the cumulative sum of the first n probabilities. Finally, roll 10 random numbers to choose 10 genes.

*3.3.3 mutation.* Mutation means that some gene values in individual coding strings are replaced by other gene values according to the mutation probability to form a new individual. Mutation operations in Genetic Algorithm are secondary methods of generating new individuals, which determine the local search capabilities of Genetic Algorithm while preserving the diversity of the population. Cross-matching and mutation operations complement each other to complete the search of the global search and local search. That to say, change the gene to adapt to the overall search, avoiding local optimal solutions. I randomly selected some genes and make these gene randomly changes.

*3.3.4 cross.* Gene cross is the replacement of two basic parts of the structure to generate new individuals. Because the crossover is best used in binary, so I write a function to convert a decimal to a binary number. And I also write a function to convert a binary number to a decimal. Randomly selected length of the A gene and randomly selected length of the B basic combination to produce a new gene.

Repeat 30 times for gene selection and cross-select an optimal gene(utility), and then bid on this utility.

4 PERFORMANCE

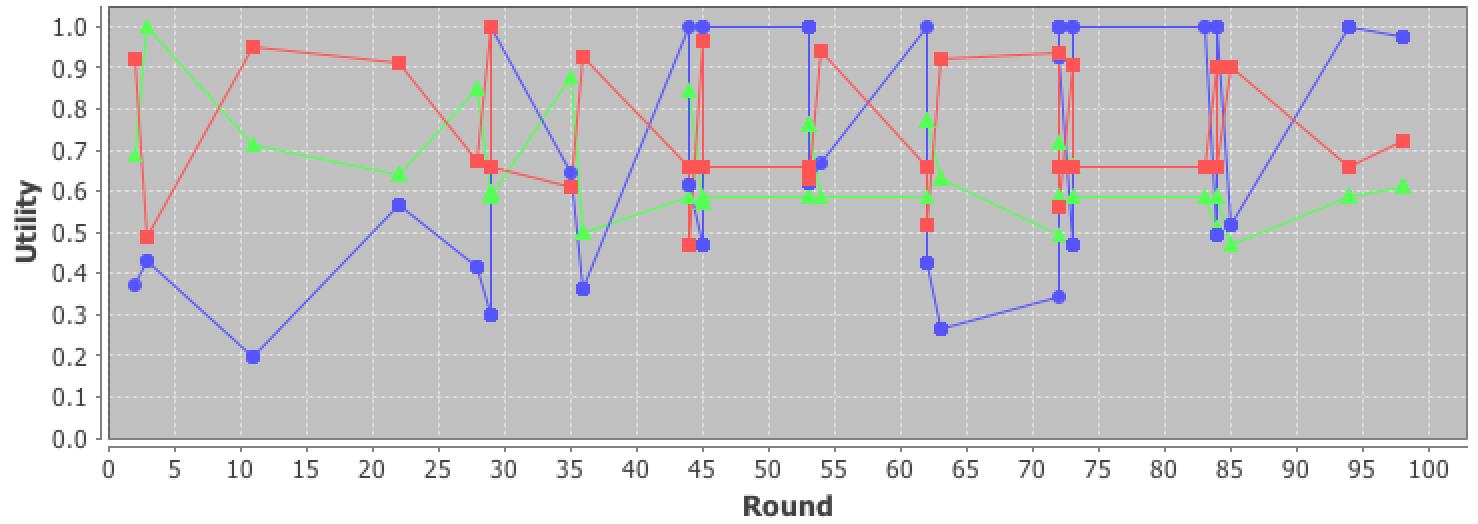


figure 1. combat result

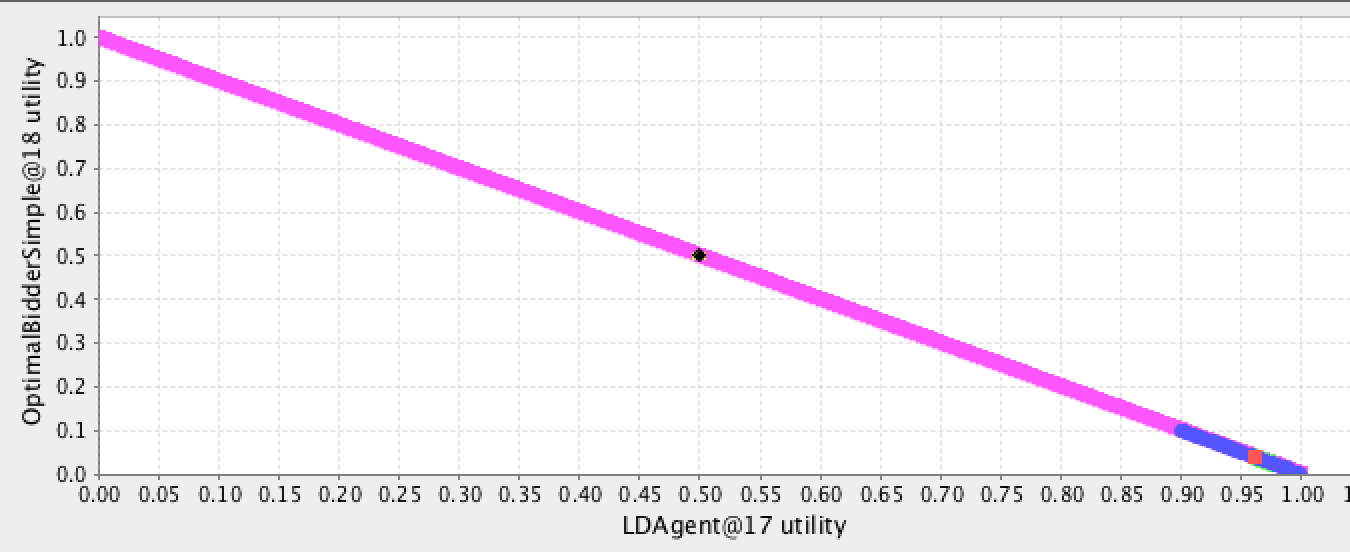
I use LDAgent, AgentLight and Atlas32016 to combat with each other.

|  |  |  |  |
| --- | --- | --- | --- |
| 1-1 | 0.9241666666666667 | 0.33721428511496104 | 0.5385685735696673 |
| 1-2 | 0.6583333333333333 | 1.0 | 0.5856226065195247 |
| 1-3 | 0.48749999999999993 | 0.43241457513409964 | 0.9999999999999999 |
| 2-1 | 0.9033333333333333 | 0.5161097156742358 | 0.46694055696422826 |
| 2-2 | 0.6583333333333333 | 1.0 | 0.5856226065195247 |
| 2-3 | 0.48749999999999993 | 0.43241457513409964 | 0.9999999999999999 |
| 3-1 | 1.0 | 0.2969260373579895 | 0.5968500104495003 |
| 3-2 | 0.6583333333333333 | 1.0 | 0.5856226065195247 |
| 3-3 | 0.5 | 0.44889131368935486 | 0.9883966566884121 |
| 4-1 | 0.9149999999999999 | 0.380054083092503 | 0.7397288204863893 |
| 4-2 | 0.6583333333333333 | 1.0 | 0.5856226065195247 |
| 4-3 | 0.5 | 0.44889131368935486 | 0.9883966566884121 |

In the first half time, the LDAgent always choose high utility of biding. But the utility value is not always 1.

|  |  |  |  |
| --- | --- | --- | --- |
| 95-1 | 0.8358333333333333 | 0.574056154481227 | 0.5831559891593853 |
| 95-2 | 0.7216666666666667 | 0.9523769815965673 | 0.6516396056070828 |
| 95-3 | 0.7216666666666667 | 0.9523769815965673 | 0.6516396056070828 |
| 96-1 | 0.9666666666666668 | 0.46848669727080305 | 0.5725677555043213 |
| 96-2 | 0.6583333333333333 | 1.0 | 0.5856226065195247 |
| 96-3 | 0.6583333333333333 | 1.0 | 0.5856226065195247 |
| 97-1 | 0.6691666666666666 | 0.5326023990508305 | 0.7102380793591041 |
| 97-2 | 0.7216666666666667 | 0.9761884907982836 | 0.6120294061545479 |
| 97-3 | 0.7216666666666667 | 0.9761884907982836 | 0.6120294061545479 |
| 98-1 | 0.7216666666666667 | 0.9761884907982836 | 0.6120294061545479 |
| 98-1 | 0.7216666666666667 | 0.9761884907982836 | 0.6120294061545479 |

This is the part of outcome, as the figure and data show below, the LDAgent get the 0.722 utility. And the distance to Nash is 0.13205. The LDAgent shows medium performance. I also use LDAgent to combat with other agent it awalys shows medium performance, not very good but not very bad. And I also change the preference profile the LDAgent also shows medium performance.



|  |  |  |
| --- | --- | --- |
| 82-1 | 0.923 | 0.077 |
| 82-2 | 0.964 | 0.036 |
| 83-1 | 0.982 | 0.018 |
| 83-2 | 0.963 | 0.037 |
| 84-1 | 0.965 | 0.035 |
| 84-2 | 0.963 | 0.037 |
| 85-1 | 0.917 | 0.083 |
| 85-2 | 0.963 | 0.037 |
| 86-1 | 0.98 | 0.02 |
| 86-2 | 0.963 | 0.037 |
| 87-1 | 0.986 | 0.014 |
| 87-2 | 0.962 | 0.038 |
| 88-1 | 0.976 | 0.024 |
| 88-2 | 0.962 | 0.038 |
| 89-1 | 0.982 | 0.018 |
| 89-2 | 0.961 | 0.039 |
| 90-1 | 0.988 | 0.012 |
| 90-2 | 0.961 | 0.039 |
| 91-1 | 0.947 | 0.053 |
| 91-2 | 0.961 | 0.039 |
| 92-1 | 0.961 | 0.039 |
| 92-1 | 0.961 | 0.039 |

The LDAgent performance well in tow agent battle, but the distance to Nash is 0.65195. However the LDAgent performance well in utility.

5 SOME PROBLEMS ENCOUNTERED

I do not know how to make a bid, it very hard to measure the distance to Nash. So I decided to use a random function to produce a random bid which the utility is between the upper limit and down limit.

The performance of the LDAgent is not very good, I want to add some analyses of opponent’s bid, but there are three agent to combat, it is not very easy for me to establish an analysis model.

I try to predict the Nash equilibrium, but it always wrong.

6 CONCLUDSION

After finishing this coursework, I know the Stacked Alternating Offers Protocol. And know how to create a automated negotiation agent. In my agent, at start time, I make the utility of bid always high, but it will not perform very well. Then I change the algorithm to make the utility linear decrease from 1 to 0, but it also not preforms very well. And then, I choose the Genetic Algorithm to make the utility to be selected from Gaussian distribution, and it performs better than before. And the most difficult part of this algorithm is the selection function and cross function, it is very hard to change a decimal to binary, because the utility is less than 1 and bigger than 0, it is not a integer. Some times when the decimal convert too many binary number, it will lead the variable overflowed. And then, I change the variable to String variable, but it still have some problems. Finally, I limit the number of bits in binary to 18 bits. After that, it can work.

REFERENCES

Jones, G., Willett, P., Glen, R., Leach, A. and Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking 1 1Edited by F. E. Cohen. Journal of Molecular Biology, 267(3), pp.727-748.

Norman, T. and Jennings, N. (2002). Constructing a virtual training laboratory using intelligent agents. International Journal of Continuing Engineering Education and Lifelong Learning, 12(1/2/3/4), p.201.

Shir, Ofer M. (2012). "Niching in Evolutionary Algorithms". In Rozenberg, Grzegorz; Bäck, Thomas; Kok, Joost N. [Handbook of Natural Computing](https://link.springer.com/referenceworkentry/10.1007/978-3-540-92910-9_32). Springer Berlin Heidelberg. pp. 1035–1069. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1007/978-3-540-92910-9\_32](https://doi.org/10.1007%2F978-3-540-92910-9_32). [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [9783540929093](https://en.wikipedia.org/wiki/Special:BookSources/9783540929093).

Taherdangkoo, Mohammad; Paziresh, Mahsa; Yazdi, Mehran; Bagheri, Mohammad Hadi (19 November 2012). "An efficient algorithm for function optimization: modified stem cells algorithm". Central European Journal of Engineering. 3 (1): 36–50. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.2478/s13531-012-0047-8](https://doi.org/10.2478%2Fs13531-012-0047-8).

LI, C., LI, G., TAN, Y. and XU, X. (2013). Medical image registration algorithm based on Powell algorithm and improved genetic algorithm. Journal of Computer Applications, 33(3), pp.640-644.