Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Comparing the Performance of Deterministic Dynamic Adaptation GA and Self Adaptive GA in Online Auctions Environment

Kim Soon, Gan
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Sabah, Malaysia
gks967@gmail.com

Patricia, Anthony
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Sabah, Malaysia
panthony@ums.edu.my

Jason, Teo
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Sabah, Malaysia
jtwteo@ums.edu.my

Kim On, Chin
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Sabah, Malaysia
kimonchin@gmail.com

Abstract— **The proliferation of online auctions has caused the increasing need to monitor and track multiple bids in multiple auctions. As a solution to the problem, an autonomous agent was developed to work in a flexible and configurable heuristic decision making framework that can tackle the problem of bidding across multiple auctions that apply different protocols (English, Vickrey and Dutch). Due to the dynamic and unpredictable nature of online auctions, the agent utilizes genetic algorithm to search for effective solution. Instead of using the conventional genetic algorithm, this paper investigates the application of deterministic dynamic adaptation genetic algorithm and self adaptive genetic algorithm to search for the most effective strategies (offline). An empirical evaluation on the comparison between the effectiveness of self-adaptive genetic algorithm and deterministic dynamic adaptation genetic algorithm for searching the most effective strategies in the online auction environment are discussed in this paper.**

Keywords-component; Online Auction; Bidding Strategies; Genetic Algorithm; Deterministic Dynamic Adaptation; Self-Adaptation

I. INTRODUCTION

Auction is defined as a bidding mechanism, described by a set of auction rules that specify how the winner is determined and how much he has to pay [1]. The used of auction can be tracked back since 500 B.C. where people used it to allocate scarce resources in Babylon [2]. Online auction is an Internet-based version of a traditional auction [3]. With online auction many of the limitations in a traditional auction are diminished such as geographical limitation, set up cost, advertising and potential bidders.

As the number of auctions increases, the process of monitoring, tracking bid and bidding in multiple auctions become a problem. Users need to monitor a number of huge auctions sites, pick the right auction to participate, and make the right bid in making sure that the desired item satisfies the user's preference. All these tasks are somewhat complex and time consuming. The task gets more complicated when there are different start and end times and when the auctions employ different protocols. For these reasons, software such as automated bidding and bid sniping are provided either by the online auction hosts or third parties to assist consumers when bidding in online auctions. However, this software has some shortcomings.

- It is only for a particular auction with a particular protocol.
- It only remains in the same auction site and does not move to other auction sites.
- It still need the intervention of the user in that the user still needs to make decision on the starting bid (initially) and the bid increments.

To address these shortcomings, an autonomous agent that can participate in multiple heterogeneous auctions, that is empowered with trading capabilities and that can make purchases autonomously was developed [4].

The bidding strategies used by the autonomous agent are heavily influenced by the value k and β of the polynomial function corresponding to each of the four constraints (the remaining time left, remaining auction left, the user's desire for bargain and the user's level of desperateness) which will be discussed in Section III where,

- k is a constant that determines the value of the starting bid
- β is the rate of concession to the private valuation

Due to the wide range of k and β values, there are infinite possibilities for the combinations of *k* and *β*. Genetic algorithm is employed to search for the approximate best bidding strategies for the auction environment. This is because genetic algorithm has shown to perform well in large search space with little priori information.

Genetic algorithm was first introduced by John Holland in 1975, and developed further by other researchers. Genetic algorithm is a search technique used in computing to search for best approximate solutions in large search space for optimization and search problems. This algorithm are a particular class of evolutionary computation that use techniques inspired by evolution biology such as inheritance, mutation, selection, and crossover (also called recombination) [5]. However, all algorithms have their limitation. Thus, various techniques have been investigated to improve the performance of genetic algorithms. One of the techniques that has shown to improve genetic algorithm performance is deterministic dynamic adaptation. Deterministic dynamic adaptation is a discipline which alters the value of strategy parameter by some deterministic rule [6]. This rule will modify the value of the strategy parameter deterministically. Normally, a time-varying schedule is used. An example of a rule is one that alters the mutation rate or crossover rate over the number generation. Self adaptation is another discipline applied in evolutionary algorithms as a method to improve the limitation in the performance of evolutionary algorithms. Self adaptation is a method to adjust the setting of the control parameters using the algorithm where the control parameters are embedded in the chromosome of the individual [7]. The principle of self-adaptation has commonly been used in evolutionary programming and evolutionary strategies but it is rarely use in genetic algorithms.

The main contribution of this paper is to make a comparison between the performances of the bidding strategies evolved from deterministic dynamic adaptation genetic algorithm and self-adaptive genetic algorithm in an online auction environment. The existing bidding strategy is evolved using a standard genetic algorithm which applied a fixed crossover and mutation rates. However, researchers have shown that varying the operator's probability is preferable than a fixed constant operator's probability. Besides that, deterministic dynamic adaptation and self-adaptation has shown to be able to improve the effectiveness of genetic algorithm [8][9][10][11][12][13]. Hence, the motivation for this work is to investigate whether using deterministic dynamic adaptation and self-adaptation can improve the bidding strategy by evolving better bidding strategies in an online auction setting.

The remainder of the paper is organized as follow. Section II reviews some of the related work. The bidding strategy framework is discussed in Section III. The experimental set up is discussed in Section IV. Section V discusses the experimental evaluation and finally Section VI presents the conclusion and future work.

II. RELATED WORK

Researchers discovered that varying the mutation probability is preferable than a fixed constant mutation probability. A study was conducted by Fogarty [8] on the dynamic mutation rate control for genetic algorithm. A deterministic schedule was used to decrease the mutation rate exponentially over the generation in this study. The result of the study showed an increase in the performance of the genetic algorithm. A similar study which took into account the string length and the population size in the decreasing mutation rate control parameter control schedule is conducted by Hesser and Manner [9][10]. Although the result of the study showed improvement in the performance, the constants c_i is difficult to be estimated in a complex problem. Besides that, Back and Schutz investigated the role of mutation in canonical genetic algorithm [11]. In this investigation, three different methods for mutation probabilities were used including fixed constant mutation probability, time dependent mutation rate schedule and self-adaptation mechanism. The result of the investigation showed that deterministic schedule performed better than the other control mechanism.

Fogel applied self-adaptation for non-numerical problems where the relative probabilities of five mutation operators are self-adapted for the components of a finite state machine [12]. Another research on the self-adapting mechanism for incremental evolution is by genetically encoding the mutation rates within a steady genetic algorithm. The experimental result showed that the self-adapting mechanism outperformed many variety of standard fixed mutation rates algorithm [13]. Hence, in this paper the performance of these two methods of adaptation are investigated to see whether they will be able to evolve better bidding strategies in an auction environment.

III. BIDDING STRATEGY FRAMEWORK

The bidding strategy framework has been developed in the previous work [4][14][15]. Before describing the decision-making framework, it is necessary to detail our assumptions about the environment.

1. Three auction protocols are considered: English, Dutch and Vickrey (three of the most common types).
2. All auctions have a known start time and English and Vickrey auctions have a known end time.
3. Our bidding agent is given a deadline (t_{max}) by when it must obtain the desired item and it is told the consumer's private valuation (p_r) for this item.
4. The agent must not buy more than one instance of the desired item.

Figure 1 shows the overview of the market simulator.

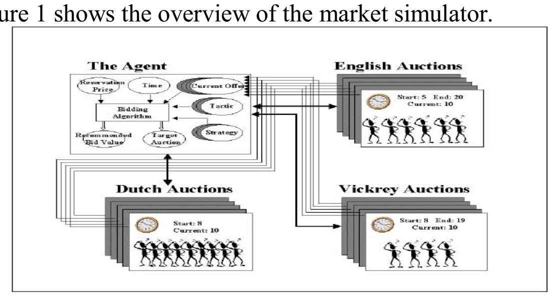

Figure 1. The market simulator

In order to bid in the auction, the agent considers several *bidding constraints* including:

- the remaining time left
- the remaining auctions left,
- the user's desire for a bargain and
- the user's level of desperateness.

For each such bidding constraint, there is a corresponding function that suggests the value to bid based on that constraint at that time. These (polynomial) functions (based on [14]) are parameterized by two key values: k (range [0..1]) is a constant that determines the value of the starting bid and β (range [0.005 - 1000]) defines the shape of the curve (and so the rate of concession to p_r). Figure 2 shows the bid values generation based on different bidding strategies. The different curves indicate the different combinations of the bidding constraints. For example (0.50, 0.00, 0.50, 0.00) implies that this strategy only values the time left, and the desire for a bargain and it sees both constraints as being equally important. All behaviors in between are also possible by setting the parameters appropriately. Based on the value of the current maximum bid, the agent selects the potential auctions in which it can bid and calculates what it should bid at this time in each such auction. The auction and corresponding bid with the highest expected utility is then selected from the potential auctions as the target auction. Finally, the agent bids in the target auction.

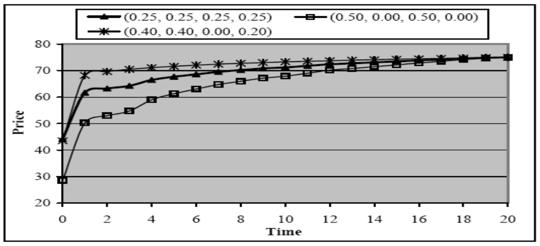

Figure 2. Various combination of bidding constraint

A series of experiments were conducted in a controlled dynamic environment to test the efficiency (in terms of success rate and average payoff) of the agent's strategy [14]. The number of strategies that can be employed is infinite because the bidding agent is heavily influenced by the strategy employed which in turn relates to the values of k and β in the given tactics and the weights for each tactic when these are to be combined. For this reason, genetic algorithm is employed to search for the approximate best bidding strategies for the auction environment. A fixed crossover rate of P_c = 0.4 and fixed mutation rate of P_m = 0.02 were used for the evolution process. However, the fixed crossover and mutation rate may not perform best in this environment. Thus, this investigation uses a non-fixed crossover and mutation rate by applying deterministic dynamic adaptation and self-adaptation to evolve the bidding strategies.

The performance of the individuals in the population is evaluated using the fitness function. The fitness function used to evaluate the performance of the individual in the population takes into account the utility of winning an auction i which is computed as

$$u_i(v) = \left(\frac{p_r - v}{p_r} \right) + c \tag{1}$$

where v is winning bid and c is arbitrary constant ranging from 0.001 to 0.005 to ensure the agent is awarded with some value in case the winning bid is equivalent to its private valuation. The individual is penalized if it fails to get the item. In this case the penalty incurred ranges from 0.01 to 0.05. These values were chosen to analyze how the population evolves with varying degrees of penalty. The fitness score is then computed by taking the average utility from a total of 2000 runs. The number of runs is fixed to 2000 times to decrease the estimated standard error of mean to a minimal level based on statistical analysis.

IV. Experimental Setting

The technique of deterministic dynamic adaptation and self-adaptation are incorporated in the genetic algorithm. The genes of the individual for deterministic dynamic adaptation contain the parameters for the four bidding tactics and the associated weight of the bidding tactics. The values for all the parameters are represented as floating points. Figure 3 shows the encoding for a bidding strategy for deterministic dynamic adaptation.

k_{rt}	β_{rt}	k_{ra}	β_{ra}	k_{ba}	β_{ba}	k_{de}	β_{de}	w_{rt}	w_{ra}	W_{ba}	w_{de}

Figure 3. Individual encoding of bidding strategy

As for the genes of the individual for self-adaptation, besides the parameters for the four bidding tactics and the associated weight of the bidding tactics the strategy parameter are also encoded into the genes of the individual representation. Instead of having a fixed global value for the crossover and mutation rate, these two values will be encoded as part of the chromosome and these values will be evolved by the evolution process same as other encoded parameters. The crossover and mutation process in self-adaptation genetic algorithm are based on the crossover and mutation rate of the parent that is selected to produce offspring. Figure 4 shows the encoding for a bidding strategy.

k_{rt}	β_{rt}	k_{ra}	β_{ra}	k_{ba}	β_{ba}	k_{de}	β_{de}	w_{rt}	w_{ra}	w_{ba}	w_{de}	p_c	p_m

Figure 4. Individual encoding of bidding strategy for self-adaptation

Table I and II show the evolutionary and parameter setting for the deterministic dynamic adaptation genetic algorithm. Figure 5 shows the deterministic dynamic adaptation genetic algorithm.

TABLE I. DETERMINISITC DYNAMIC ADAPTATION EVOLUTIONARY SETTING

Representation	: Real Values Number
Crossover	: Extension Combination Operator
Mutation	: Creep Operator
Selection	: Tournament Selection

TABLE II. DETERMINISTIC DYNAMIC ADAPTATION PARAMETER SETTING

Number of Generations	50
Number of Individuals	50
Elitism	10%
Crossover Probability	Change(Range from 0.4 to 0.6)
Mutation Probability	Change (Range from 0.2 to 0.002)
Termination Criteria	After 50 Generation
Numbers of Repeat Run	30

```
Begin
   Randomly create initial bidder populations;
   While not (Stopping Criterion) do
     Calculate fitness of each individual by running the
     marketplace 2000 times;
     Create new population
       Select the fittest individuals (HP);
       Create mating pool for the remaining population;
       Perform crossover and mutation in the mating
       pool to create new generation(SF);
       New generation is HP + SF;
   Change The Control Parameters Values.
   Gen = Gen + 1
   End while
End
```

Figure 5. Deterministic Dynamic Adaptation Genetic Algorithm

There are eight different variation schemes that are taken into account in this work. The schemes are shown in Table III. The deterministic increasing scheme for the crossover rate will change progressively from p_c = 0.2 to p_c = 0.6 over the generation whereas the decrease scheme for the crossover rate is the opposite of the increasing scheme, which is from p_c = 0.6 to p_c = 0.2. The deterministic increasing scheme for the mutation rate, changes the rate progressively from p_m = 0.002 to p_m = 0.2 over the generation while the deterministic decreasing scheme is the opposite of this. The range for crossover rate in deterministic dynamic adaptation is selected by taking the 0.4 as the midpoint [16]. The range for mutation rate in deterministic dynamic adaptation is also selected by taking the 0.02 as the midpoint [17].

TABLE III. DIFFERENT DETERMINISTIC DYNAMIC ADAPTATION SCHEMES

Crossover	Mutation	Abbreviation
Increase	Fixed	CIMF
Fixed	Increase	CFMI
Decrease	Fixed	CDMF
Fixed	Decrease	CFMD
Decrease	Decrease	CDMD
Decrease	Increase	CDMI
Increase	Decrease	CIMD
Increase	Increase	CIMI

The evolutionary setting for self-adaptation is shown in Table IV. The algorithm for the self-adaptation genetic algorithm is shown in Figure 6. The difference between self-adaptation GA and the traditional GA is the crossover and mutation process is also applied to the crossover probability and the mutation probability which encoded in the representation.

TABLE IV. DETERMINISTIC DYNAMIC ADAPTATION PARAMETER SETTING

Number of Generations	50
Number of Individuals	50
Elitism	10%
Crossover Probability	Self-adapted by the algorithm
Mutation Probability	Self-adapted by the algorithm
Termination Criteria	After 50 Generation
Numbers of Repeat Run	30

```
Begin
   Randomly create initial bidder populations;
   While not (Stopping Criterion) do
        Calculate fitness of each individual by running the
        marketplace 2000 times;
        Create new population
           Select the fittest individuals (HP);
           Create mating pool for the remaining population;
           Perform crossover and mutation in the mating
           pool to create new generation(SF);
           New generation is HP + SF;
         Gen = Gen + 1
   End while
End
```

Figure 6. Self-Adaptive Genetic Algorithm

The crossover process used in the evolution process is an extension combination operator with two crossover points. Two individuals are randomly selected from the mating pool and using 2 crossover points that are randomly picked to exchange their genetic material. The exchanging of genetic material process is performed using an extension combination operator [18][19], which works by taking the difference between two values of the crossover point, adding this difference to the higher (giving the maximum range) and subtracting it from the lower (giving a minimum range). The new values are then generated between the minimum and maximum range. The creep operator is used for mutation to allow a small value to be subtracted from the gene, but instead of adding or subtracting a random number, a small constant 0.05 is used [18][19]. The gene from the chosen individual is picked randomly and 0.05 is added or subtracted, depending on the range limitation for that particular gene. The mutation process is only applied to the values of k and β for each tactic. The weights are not considered here because adding a small value to the weight requires a renormalization and will have very little effect on the agent's online behavior.

V. EXPERIMENTAL EVALUATION

The purpose of the experiment is to compare the performances of the bidding strategies evolved from different schemes of deterministic adaptation in genetic algorithm and the self-adaptation genetic algorithm.

First, we evolved the strategies using the mutation rate p_m = 0.02 and the two deterministic crossover control rule. Then, we evolved the strategies using the crossover rate p_c = 0.4 and the two deterministic mutation control rule. Secondly, we evolved the strategies with self-adaptation genetic algorithm, where the crossover and mutation are generated randomly based on Gaussian distribution.

The parameters setting for the simulated electronic market place for the empirical evaluations are shown in Table V. The parameters include the agent's reservation price, the agent's bidding time and the number of active auctions. The agent reservation price is the maximum amount that the agent is willing to pay for the item. The bidding time is the time allocated for the agent to obtain the user's required item. The active auctions are the list of auctions that is ongoing before time t_{max}.

TABLE V. CONFIGURABLE PARAMETERS FOR THE TESTING ENVIRONMENT

Agent reservation price	$73 \leq p_r \leq 79$
Bidding time for each auction	$21 \leq t_{max} \leq 50$
Number of active auction	$20 \leq L(t) \leq 45$

The performance of the evolved strategies is evaluated based on three measurements. Firstly, the average fitness is the fitness of the population at each generation over 50 generations. The average fitness shows how well the strategy converges over time to find the best solution.

Secondly, success rate is the percentage of time that an agent succeeds in acquiring the item by the given time at any price less than or equal to its private valuation. This measure will determine the efficiency of the agent in terms of guaranteeing the delivery of the requested item. Finally, the third measurement is the average payoff which is defined as

$$\frac{\sum_{1 \le x \le 100} \left(\frac{p_r - v_i}{p_r} \right)}{n} \tag{3}$$

where p_r is the agent's private valuation, n is the number of runs, v_i is the winning bid value for auction i. This value is then divided by the agent's private valuation, summed and average over the number of runs. The agent's payoff is 0 if it is not successful in obtaining the item.

VI. Result and Discussion

This paper conducted two set of experiments whereby the first experiment compares the different deterministic dynamic adaptation schemes and the second experiment compares the self-adaptive GA with the best deterministic dynamic adaptation schemes. The purpose of doing it this way is to investigate which genetic algorithm disciplines can perform better in evolving the auction bidding strategies. Figure 7 show that the population with CDMI achieving a higher average fitness when compared to the other deterministic adaptation schemes. This scheme is superior as it is able to maintain sufficient diversity of the population till the end of the run and the exploitation is best in this genetic algorithm environment compared to the others. This is an indication that the population evolved from the CDMI will have higher fitness and would most likely perform better than the individuals from the rest of mutation rate. The next two experiments will be able to confirm this hypothesis.

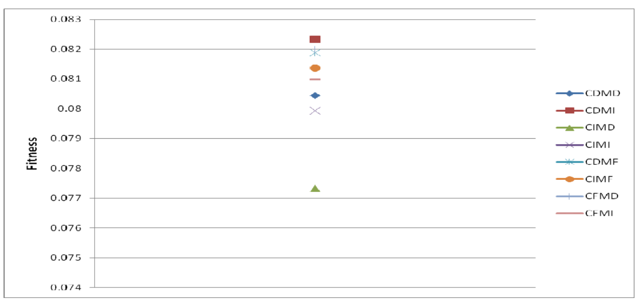

Figure 7. Average fitness of population for different deterministic adaptation genetic algorithm schemes

The success rate and the average payoff are important measurements for the evolving strategies. 10 individuals evolved from the each scheme population are selected and are run in our simulated auction market. For each strategy, we ran it 200 times in the marketplace. Figure 8 shows the comparison of the success rate between the individuals of the testing set. The result shows that the strategy evolved using CDMI performs the best overall. The individuals evolved from the CDMI outperformed the individuals from the other control rules schemes by delivering a 1.5% to 2.5% increase in success rate compared to the individuals evolved from the other deterministic schemes. Here, we can conclude that the strategy evolved from the CDMI can evolve better strategies and delivers higher success rate when bidding in online auctions thus improving the GA in searching for better bidding strategies.

Figure 9 shows the individuals evolved from the CDMI outperformed the rest with a higher average payoff of between 2% and 4%. This shows that the strategy evolved using the CDMI does not only generate a better average fitness but also evolves better effective strategies compared to the strategy evolved for the other seven deterministic adaptation schemes. The result shown here confirms the hypothesis we have made earlier, where individuals in population with a better average fitness would performed best. Based on the results of the experiments that had been carried out, it can be concluded that the strategies evolved with CDMI performed better than the strategies evolved out of the other seven schemes in terms of success rate and average payoff in an online auction environment setting.

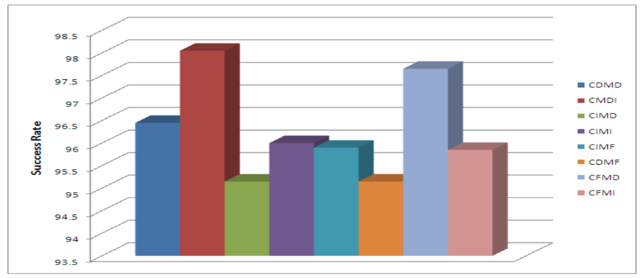

Figure 8. Success rate for strategies evolved from different deterministic adaptation genetic algorithm schemes

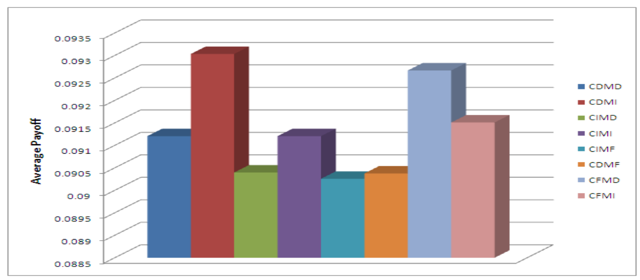

Figure 9. Average Payoff for strategies evolved from different deterministic adaptation genetic algorithm schemes

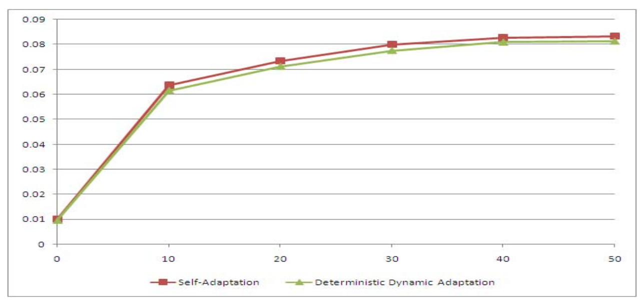

Figure 10. Average fitness of population for deterministic adaptation GA and self-adaptation GA

Figure 10 shows the average fitness for the evolving bidding strategy from deterministic adaptation (CDMI) GA and self-adaptation GA. The CDMI scheme is selected to be compared with the self-adaptive GA because it's the best performer in the deterministic dynamic adaptation experiment. The population with self adaptation GA achieved a higher average fitness when compared to the CDMI. Self adaptive GA performed better because the nature of the self-adaptation is to focus on the appropriate region of the search space thus allowing better bidding strategies to be found.

The result from Figures 11 and 12 show the success rate and the average payoff for the strategies evolved using self-adaptation genetic algorithm and deterministic adaptation GA. Self-adaptation GA performs the best overall compared to deterministic genetic algorithm as predicted. The results obtained here indicate that the bidding strategies evolved from self-adaptive genetic algorithm performed better than deterministic genetic algorithm in the online auction environment. This is due to its self-adaptive nature which is able maintains sufficient diversity among individuals in order to enable further evolvability.

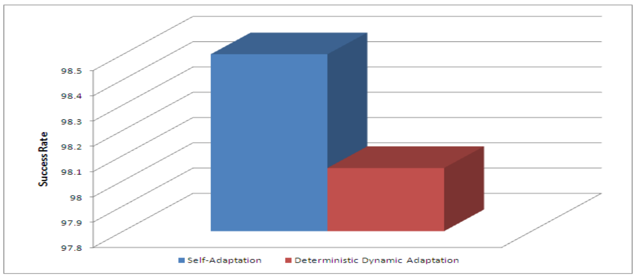

Figure 11. Success rate for bidding strategies evolved using CFMD and self-adaptation GA

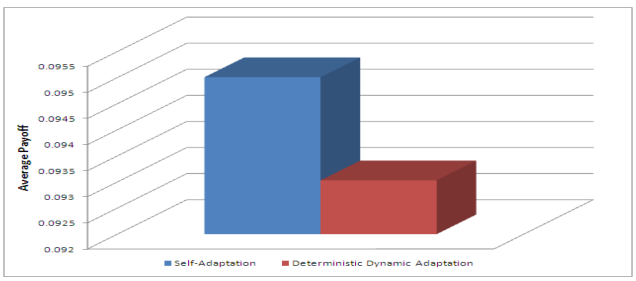

Figure 12. Average payoff for bidding strategies evolved using CFMD and self-adaptation GA

VII. Conclusion

This paper investigates the relative performance of the different deterministic adaptation genetic algorithms and self-adaptation genetic algorithm on a flexible and configurable heuristic decision making framework that can tackle the problem of bidding across multiple auctions that apply different protocols (English, Vickrey and Dutch). The polynomial nature of the strategies formulation creates a large search space making it difficult to search for the optimal solution. In this paper, eight different deterministic adaptation control rules have been applied to the GA to search for the effective strategy. The best deterministic adaptation scheme (in this case, CDMI) is then compared with the self-adaptive genetic algorithm. The experimental evaluation showed that the strategies evolved from the dynamic decrease mutation rate performed better than the other strategies evolved from the other dynamic adaptation schemes in terms of success rate and average payoff when bidding in the online auction marketplace. However, self-adaptive genetic algorithm is able to outperform the best performer for the deterministic adaptation scheme. There are several areas that require further investigation. First, further exploration can be conducted on how well genetic algorithms can help in improving the performance of the bidding strategies by using different discipline of evolutionary algorithm such as evolution strategies.

References

[1] Wolfstetter, E., Auctions: An Introduction. *Journal of Economic Surveys 10*, 367-420, 2002.

[2] Shubik, M., *Auctions, Biddings, and Markets: An Historical Sketch.* New York, USA: New York University Press, 1983.

[3] Jansen, E., Netlingo the Internet Dictionary. http://www.netlingo.com/. 2003.

[4] Anthony, P., Jennings, N.R.: *Agent for Participating in Multiple Online Auctions*. ACM Transaction on Internet Technology 3(3), 1–32 (2003)

[5] Engelbrecht, A.P., *Computational Intelligence An Introduction*. John Wiley & Sons, USA, 2002.

[6] Hinterding, R., Michalewicz, Z. and Eiben, A.E. *Adaptation in Evolutionary Computation : A Survey*, Proceedings of The IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, 1997.

[7] Meyer-Nieberg, S. and Beyer, H-G., Self-Adaptation in Evolutionary Algorithms. In F. Lobo, C. Lima, and Z. Michalewicz, editors, *Parameter Setting in Evolutionary Algorithm,* Springer, Berlin, 2006.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[8] Fogarty T., Varying the probability of mutation in genetic algorithm, Proc. of the Third International Conference on Genetic Algorithms, pp. 104-109, Morgan Kaufmann, 1989.

[9] Hesser J. and Manner R., Towards an Optimal Mutation Probability in Genetic Algorithms, Proc. of the 1st Parallel Problem Solving from Nature, pp. 115-124, Springer, 1991.

[10] Hesser J. and Manner R., Investigation of the m-heuristic for optimal mutation probabilities, Proc. of the 2nd Parallel Problem Solving from Nature, pp.115-124, Elsevier, 1992.

[11] Back T. and Schutz M., Intelligent Mutation Rate Control in Canonical Genetic Algorithms, Proc. of the International Symposium on Methodologies for Intelligent Systems, pp. 158-167, 1996.

[12] Lawrence J. Fogel, Peter J. Angeline, David B. Fogel: An Evolutionary Programming Approach to Self-Adaptation on Finite State Machines. In J.R. McDonell, R.G. Reynolds, and D.B. Fogel, editors, Proceedings of the Forth Annual Conference on Evolutionary Programming, Massachusetts, 1995. MIT Press. pp. 355-365.

[13] Smith, J.; Fogarty, T.C., "Self adaptation of mutation rates in a steady state genetic algorithm," *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on* , vol., no., pp.318-323, 20-22 May 1996

[14] Anthony,, P. and Jennings, N. R., Evolving bidding strategies for multiple auctions. *In Proceedings of the 15th European Conference on Artificial Intelligence*. 178–182. 2002

[15] Anthony, P., *Bidding Agents for Multiple Heterogeneous Online Auctions.* PhD thesis, University of Southampton, UK, 2003.

[16] Gan K.S., Anthony P. and Teo J. 2008. The Effect of Varying the Crossover Rate in the Evolution of Bidding Strategies. 4th International IASTED Conference on Advances in Computer Science and Technology (ACST-2008), Langkawi, Malaysia, April 2008

[17] Gan K.S., Anthony P. and Teo J. 2008, Mutation Rate in The Evolution of Bidding Strategies, The 3rd International Symposium on Information Technology 2008 (ITSim2008), Kuala Lumpur, Malaysia, August 2008I.

[18] D. Beasley, D. R. Bull, and R. R. Martin (1993a). An Overview of Genetic Algorithms: Part 1, Fundamentals. University Computing 15 (2), 58-69

[19] D. Beasley, D. R. Bull, and R. R. Martin (1993a). An Overview of Genetic Algorithms: Part 2, Research Topics. *University Computing 15* (4), 170 – 181.