2016

A Genetic Algorithmic Approach to Automated Auction Mechanism Design

Jinzhong Niu
CUNY Guttman Community College

Simon Parsons
King's College London

A Genetic Algorithmic Approach to Automated Auction Mechanism Design

Jinzhong Niu[1] and Simon Parsons[2]

[1] Guttman Community College, City University of New York, USA
jniu@gradcenter.cuny.edu
[2] Department of Informatics, King's College London, UK
simon.parsons@kcl.ac.uk

Abstract. In this paper, we present a genetic algorithmic approach to automated auction mechanism design in the context of CAT games. This is a follow-up to one piece of our prior work in the domain, the reinforcement learning-based grey-box approach [13]. Our experiments show that given the same search space the grey-box approach is able to produce better auction mechanisms than the genetic algorithmic approach. The comparison can also shed light on the design and evaluation of similar search solutions to other domain problems.

Keywords: Genetic algorithms, auction mechanism design, double auctions, JCAT

1 Introduction

Auction mechanisms play an essential role in electronic commerce and in market-based control and resource allocation in computer systems. A major challenge in these domains is to design auction mechanisms that exhibit desired properties. Automated mechanism design aims to solve the problem of mechanism design in an automated fashion, typically by searching some space of possible mechanisms [3,5,19].

One piece of our prior work in this area is [13], in which we presented a what we called *grey-box* approach to automated design of double auctions in the context of TAC Market Design Competition (or the CAT Game) [14]. In the grey-box method, we use a tree model to represent the search space and associate an *n*-armed bandit problem solver [22, Chapter 2] to each node where multiple partial solutions to the same part of the problem exist. The *n*-armed bandit problem solvers select building blocks so that complete auction mechanisms can be constructed and evaluated in CAT games. The performance of each sampled auction mechanism in CAT games is then used as feedback for those building blocks in the mechanism. Our experiments showed that the grey-box search was able to produce better auction mechanisms than those manually crafted by participants in the first CAT Game.

As the tree model is independent of search methods, one follow-up question that arises naturally is: *How would the grey-box method perform compared to other search methods?* Indeed, other search methods have been used in automated auction mechanism design, though focusing on some particular aspects of an auction mechanism. For instance, Cliff [3] used a simple *genetic algorithm* (GA) to explore a continuum of probabilities of the next shout in an auction coming from a seller (or a buyer) and Phelps *et* *al.* [19] used *genetic programming* (GP) in acquiring pricing rules for double auctions. In this paper, aiming to answer the question raised above, we investigate how effective simple GAs are in automated auction mechanism design based on the same tree model (search space) as used in the grey-box search.

We first briefly review the grey-box method as well as the search space of double auctions in the domain of CAT games in Section 2. Due to the space constraint, detailed descriptions of these and other background information are not included here but can be found in [13].[3] Then Section 3 introduces the GA search method and Section 4 describes the GA experiments we carried out and interprets the experimental results. Section 5 further draws contrasts and makes connections between the grey-box method and various evolutionary computational solution concepts and techniques, and finally concludes.

2 A Brief Review of the Grey-Box Method

The grey-box method aimed to search for auction mechanisms in the domain of the CAT Game, an annual event held from 2007 through 2011 to foster research in electronic market mechanism design [14]. In a CAT game, participants each operate an electronic double-auction marketplace and the marketplaces compete against each other for market share and profit. Traders are software agents provided by the game organizers. Each trader is armed with a marketplace selection strategy as well as a bidding strategy so that the trader can choose a marketplace to bid and trade in. A CAT game lasts a certain number of trading days. Each trader has a chance before the start of each day to select a marketplace to trade during that day and the marketplace can impose various charges on traders, admission, transaction fee, etc. At the end of the day, each marketplace receives a daily score between 0.0 and 1.0, a combination of three components with equal weights: share of trader population attracted, share of profit, and percentage of successful trade offers. The marketplace, or indeed the auction mechanism designed for the marketplace, that receives the highest cumulative score wins the game. The execution of CAT games is supported by JCAT [15], the open source software package that we also used to run the grey-box experiments [13] and the GA experiments in this paper.

In the grey-box method, the search space is modeled as a tree, which is depicted in Fig. 1, an abbreviated version of Fig. 1 in [13]. The tree model illustrates how building blocks are selected and assembled level by level. There are and nodes, or nodes, and leaf nodes in the tree. An and node combines a set of building blocks, each represented by one of its child nodes, to form a compound building block. The root node, for example, is an and node assembling *policies*, one on each major aspect of an auction mechanism (M for matching policy, Q for quoting policy, A for shout accepting policy, C for clearing condition, P for pricing policy, and G for charging policy),[4] to construct an auction mechanism. An or node represents the decision making of selecting a building block from the candidates represented by the child nodes of the or node. A leaf node represents an atomic block that can either be for selection at its or parent node or be further assembled into a bigger block by its and parent node. A special type of leaf

[3] [13] is available at `http://www.sci.brooklyn.cuny.edu/~jniu/research/publications/`.

[4] A taxonomy of policies in this domain of auction mechanisms is described in detail in [16].

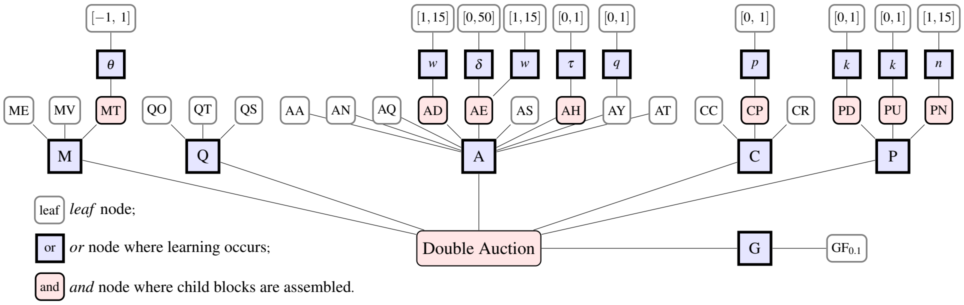


Fig. 1: The search space of double auctions modeled as a tree.

node in Fig. 1 is that with a label in the format of $[x,y]$. Such a $[x,y]$ node is a convenient representation of a set of leaf nodes that have a common parent—the parent of this special leaf node—and take values evenly distributed between x and y for the parameter labeled at the parent node. Note that both the grey-box search and the GA search to be introduced below consider only mechanisms using a fixed charging policy, denoted as GF$_{0.1}$. This simplification aims to avoid the slow exploration in the particular corner of the search space for charging policy, which involves significantly more parameters and variations than those for other policies.

The grey-box method combines techniques from reinforcement learning, e.g., solutions to n-armed bandit problem [22], and evolutionary computation, e.g., the use of a Hall of Fame [20]. The general idea of this algorithm is to use n-armed bandit learners to choose building blocks when needed so as to construct auction mechanisms based on the tree model in Fig. 1, to run CAT games to evaluate the constructed mechanisms, and to keep good mechanisms in a Hall of Fame.

In the tree model, or nodes contribute to the variety of auction mechanisms in the search space and are where exploitation and exploration occur. We model each or node as an n-armed bandit learner that chooses among candidate blocks, and we use the simple softmax method [22, Section 2.3] to solve this learning problem. Solving all the n-armed bandit learners in the tree will uniquely determine a configuration of an auction mechanism, which is exactly how an auction mechanism is sampled in the search space. The sampled mechanisms can then be put into a CAT game for evaluation. The game score of a sampled mechanism not only suggests how good the mechanism itself is, but is also an indicator of the performance of the building blocks that are used in the mechanism. If a building block is due to the selection of an n-armed bandit learner among the child nodes of the corresponding or node, the game score can be readily used as the feedback for the building block. All such feedback to a building block cumulatively serves as the expected return, or what we call the *quality score*, of the building block. Thus, after a game completes, the quality scores of building blocks that are children of an or node are updated, and so are the way how an auction mechanism is sampled in the space in later steps.

In each CAT game that is run to evaluate sampled mechanisms, we include four fixed, well known, mechanisms plus selected mechanisms that performed well at previous steps and are from the Hall of Fame. The fixed set of four mechanisms in every CAT game includes two clearing house (CH) mechanisms—CH$_l$ and CH$_h$—and two continuous double auction (CDA) mechanisms—CDA$_l$ and CDA$_h$—with one of each adopting the GF$_{0.1}$ policy, charging a *low* 10% fee on trader profit and the other charging a *high* 100% fee on trader profit. The CH and CDA mechanisms have been used in the real world for many years and were found competitive in the context of CAT games as well. The selection of Hall of Famers to compete in the CAT game is based on the same softmax method as used in choosing building blocks at each $\boxed{\text{or}}$ node. More details on how the Hall of Fame is maintained and Hall of Famers are selected can be found in [13].

3 The Genetic Algorithmic Approach

To compare the effectiveness of the grey-box approach with other search methods, we carried out a new set of experiments, searching the same solution space as used in the grey-box experiments based on the classic GA [6,10].

In these GA experiments, each individual auction mechanism is not represented by a binary string as in a typical GA, but by a tree structure, since each individual auction mechanism can be viewed as the result of making selections at the $\boxed{\text{or}}$ nodes in the tree model in Fig. 1 (it is exactly the case in the grey-box experiments), and thus be conveniently represented by the tree structure after the unselected branches of the $\boxed{\text{or}}$ nodes are cut off from the tree model. For example, the tree on the left side in Fig. 2 represents the auction mechanism ME + QS + AD$_{w=3}$ + CP$_{p=0.4}$ + PU$_{k=0.7}$ + GF$_{0.1}$.

The tree-based encoding of an individual requires specialized mutation and crossover operators, due to the hierarchical construction and the different types of node in the tree. The diversity of auction mechanism individuals in the space originates from the $\boxed{\text{or}}$ nodes, so mutation and crossover occur only at $\boxed{\text{or}}$ nodes. To apply mutation to an individual, it is decided probabilistically, based on the *mutation rate*, at each $\boxed{\text{or}}$ node in its tree-based encoding whether the node selects a different child node from the tree model. If yes, the original child (and its children if any) is replaced by the new child, which is uniformly selected from all the possible choices other than the original one. If the new child requires its own descendants, the whole subtree is added. Descendants that are $\boxed{\text{or}}$ nodes make their selections randomly, in contrast to the way in the grey-box experiments where selections are made based on the quality scores of different choices. Fig. 2 demonstrates an example of mutation on the auction mechanism given above, with the encoding before mutation on the left side and the encoding after mutation on the right side. The node C is the only place where mutation occurs and as a result the branch CP$_{p=0.4}$ is replaced by CR, both enclosed by dotted lines.

Crossover occurs between two auction mechanism individuals in the GA experiments, and only at $\boxed{\text{or}}$ nodes similar to what happens with mutation. To perform crossover, indeed single-point crossover, between two individuals, the $\boxed{\text{or}}$ nodes that appear in both trees and have different children respectively in the two trees are collected; then one of these collected nodes is selected randomly as the place to possibly perform the crossover; and finally it is decided probabilistically, based on the *crossover rate*, whether or not to perform the crossover, and if yes, the two appearances of the selected node in the two trees switch their children. Fig. 3 demonstrates the crossover

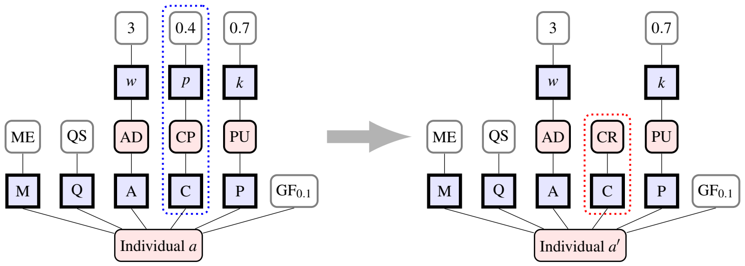


Fig. 2: An illustration of mutation in the GA, before mutation on the left side and after mutation on the right side. The replaced and replacing subtrees are both enclosed by dotted lines.

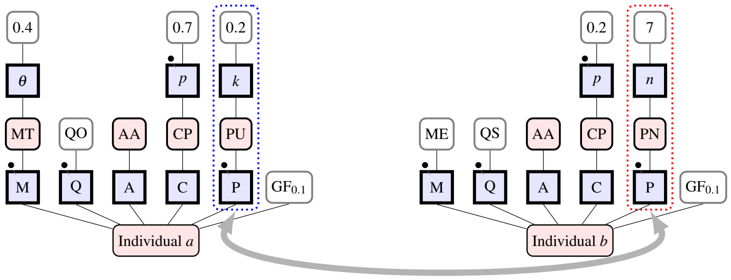


Fig. 3: An illustration of crossover in the GA between two individuals. The $\boxed{\text{or}}$ nodes at which crossover can be performed are marked with • in the original encodings. P is selected to be the place where crossover is actually performed. The two subtrees with P as the root in the two trees are swapped and enclosed by dotted lines.

between two individuals—identified as *a* and *b* in the figure respectively. In Fig. 3, the $\boxed{\text{or}}$ nodes at which crossover can be performed are marked with •, including M, Q, P, and *p*. A and C are excluded because their children in the two trees respectively are also identical, while θ and *p* in individual *a* and *n* in individual *b* are excluded because they appear in only one of the two trees. Random selection among the eligible nodes picks P. After a probabilistic test based on the crossover rate is taken and turns out to be positive, the subtrees $\mathrm{PU}_{k=0.2}$ in *a* and $\mathrm{PN}_{n=7}$ in *b*, both enclosed by dotted lines in the figure, are swapped, producing two new individuals.

The skeleton of the GA algorithm that is used in our GA experiments is given in Algorithm 1. These GA experiments adopt the same search space of auction mechanisms, the same set of fixed auction mechanisms to evaluate the fitnesses of the mechanisms sampled from the space, and the same idea of using a Hall of Fame to produce output as in the grey-box experiments.

The initial generation of auction mechanism individuals in each GA experiment is created by randomly sampling the search space in exactly the same way as at the beginning of the grey-box search until a certain number (SIZE_OF_POPULATION) of indi-

Algorithm 1: The GA-AMD algorithm.

```
Input: $\mathbb{B}$, $\mathbb{FM}$
Output: $\mathbb{HOF}$
begin
    $\mathbb{HOF} \leftarrow \varnothing$
    for $g \leftarrow 1$ to NUM_OF_GENERATIONS do
        if $g = 1$ then
            $\mathbb{P} \leftarrow$ Init-Population($\mathbb{B}$)
        else
            $\mathbb{P} \leftarrow$ Select-Population($\mathbb{P}$)
            $\mathbb{P} \leftarrow$ Crossover-Population($\mathbb{B}$, $\mathbb{P}$, $r_{co}$)
            $\mathbb{P} \leftarrow$ Mutate-Population($\mathbb{B}$, $\mathbb{P}$, $r_m$)
        $\mathbb{P} \leftarrow$ Randomize($\mathbb{P}$)
        for $i \leftarrow 1$ to $|\mathbb{P}|/$NUM_OF_SAMPLES do
            $G \leftarrow$ Create-Game()
            $\mathbb{SM} \leftarrow \varnothing$
            for $m \leftarrow 1$ to NUM_OF_SAMPLES do
                $\mathbb{SM} \leftarrow \mathbb{SM} \cup \{\mathbb{P}[(i-1) *$ NUM_OF_SAMPLES $+ m]\}$
            $\mathbb{EM} \leftarrow$ Select($\mathbb{HOF}$, NUM_OF_HOF_SAMPLES)
            Run-Game($G$, $\mathbb{FM} \cup \mathbb{EM} \cup \mathbb{SM}$)
            foreach $M \in \mathbb{EM} \cup \mathbb{SM}$ do
                Update-Market-Score($M$, Score($G$, $M$))
                if $M \notin \mathbb{HOF}$ then
                    $\mathbb{HOF} \leftarrow \mathbb{HOF} \cup \{M\}$
                if CAPACITY_OF_HOF $< |\mathbb{HOF}|$ then
                    $\mathbb{HOF} \leftarrow \mathbb{HOF} - \{$Worst-Market($\mathbb{HOF}$)$\}$
```

Function Init-Population.

```
Input: $\mathbb{B}$
Output: $\mathbb{P}$
begin
    $\mathbb{P} \leftarrow \varnothing$
    for $i \leftarrow 1$ to SIZE_OF_POPULATION do
        $M \leftarrow$ Create-Market()
        for $t \leftarrow 1$ to NUM_OF_POLICYTYPES do
            $B \leftarrow$ Select($\mathbb{B}_t$, 1)
            Add-Block($M$, $B$)
        $\mathbb{P} \leftarrow \mathbb{P} \cup \{M\}$
```

viduals are obtained (see Function Init-Population). Each of the subsequent generations is created through steps of selection, crossover, and mutation from the previous generation. The selection step, shown in Function Select-Population, is a combination of *elitism* and *roulette wheel selection*. Elitism selection keeps a certain number of fitter individuals in the next generation based on the *elitism rate*, which determines the size of the portion of the population to be considered as elite individuals. Roulette wheel selection fills the rest of the population by probabilistically selecting among all the in-

Function Select-Population.

```
Input: $\mathbb{P}$
Output: $\mathbb{P}'$
begin
    $\mathbb{P}' \leftarrow \varnothing$
    Descending-Sort($\mathbb{P}$)
    $n_e \leftarrow$ SIZE_OF_POPULATION $* r_e$
    for $i \leftarrow 1$ to $n_e$ do
        $\mathbb{P}' \leftarrow \mathbb{P}' \cup \{\mathbb{P}[i]\}$
    $s \leftarrow 0$
    for $i \leftarrow 1$ to SIZE_OF_POPULATION do
        $s \leftarrow s +$ Score($\mathbb{P}[i]$)
    for $i \leftarrow n_e$ to SIZE_OF_POPULATION do
        $k \leftarrow$ SIZE_OF_POPULATION
        $r \leftarrow$ Uniform($0,s$)
        for $j \leftarrow 1$ to SIZE_OF_POPULATION do
            $r \leftarrow r -$ Score($\mathbb{P}[i]$)
            if $r <= 0$ then
                $k \leftarrow j$
                break
        $\mathbb{P}' \leftarrow \mathbb{P}' \cup \{\mathbb{P}[k]\}$
```

Function Crossover-Population.

```
Input: $\mathbb{B}$, $\mathbb{P}$
Output: $\mathbb{P}'$
begin
    $\mathbb{P}' \leftarrow \varnothing$
    $n_e \leftarrow$ SIZE_OF_POPULATION $* r_e$
    for $i \leftarrow 1$ to $n_e$ do
        $\mathbb{P}' \leftarrow \mathbb{P}' \cup \{\mathbb{P}[i]\}$
    for $i \leftarrow 1$ to (SIZE_OF_POPULATION $- n_e)/2$ do
        $\mathbb{P}' \leftarrow \mathbb{P}' \cup \{$Crossover-Individuals($\mathbb{B}$, $\mathbb{P}[n_e + i * 2 - 1]$, $\mathbb{P}[n_e + i * 2]$,$r_{co}$)$\}$
```

Function Mutate-Population.

```
Input: $\mathbb{B}$, $\mathbb{P}$
Output: $\mathbb{P}'$
begin
    $\mathbb{P}' \leftarrow \varnothing$
    $n_e \leftarrow$ SIZE_OF_POPULATION $* r_e$
    for $i \leftarrow 1$ to $n_e$ do
        $\mathbb{P}' \leftarrow \mathbb{P}' \cup \{\mathbb{P}[i]\}$
    for $i \leftarrow n_e$ to SIZE_OF_POPULATION do
        $\mathbb{P}' \leftarrow \mathbb{P}' \cup \{$Mutate-Individual($\mathbb{B}$, $\mathbb{P}[i]$, $r_m$)$\}$
```

dividuals in the previous generation. The probability of an individual being selected each time is proportional to its fitness, which is its average daily score in the game that it participated in during the evaluation of the previous generation. This type of selection has a known problem that individuals with low fitnesses have little chance to get selected when the fitnesses of individuals differ dramatically. Due to the scoring scheme of the CAT game, the typical daily score of an auction mechanism ranges from 0.1 to 0.5, so the usual drawback of roulette wheel selection does not have big impact in this GA algorithm. The individuals that are picked in roulette wheel selection then go through the crossover and mutation steps. In the crossover step, shown in Function Crossover-Population, individuals are paired up and each pair is probabilistically recombined (`Crossover-Individuals()` in Line 7) as we described above and illustrated in Fig. 3. In the mutation step, shown in Function Mutate-Population, individuals are each probabilistically mutated (`Mutate-Individual()` in Line 7) as we described above and illustrated in Fig. 2.

To evaluate a generation of auction mechanism individuals, all the mechanisms are randomly divided into groups. For each group, a CAT game is created, and, similar to those games in the grey-box experiments, this CAT game also includes a set of fixed market mechanisms and a certain number of mechanisms sampled from the Hall of Fame. After the game, the Hall of Fame is updated to incorporate the scores of the participating Hall of Famers and include new individuals from the generation that performed well. The way in which the Hall of Fame is manipulated is exactly the same as in the grey-box experiments. As mentioned above, the average daily scores of the individuals are used as their fitnesses in the selection step.

4 Experimental Setup and Results

In the GA experiments, each game is configured to evaluate two individuals from the population as in the grey-box experiments. To compare the performances of the two approaches, the population consists of 20 individual auction mechanisms at each generation and evolves over 20 generations so that each GA experiment makes use of approximately the same number of CAT games in total (200) as in a grey-box experiment.[5] Some experiments based on the GA may have a population of thousands of individuals or even more. Our experiment cannot support a population of this size due to the high computational cost of running CAT games. The 20 generations and the population of 20 individuals are the result of balancing the two parameters under the constraint of the total number of CAT games to run. The elitism rate, r_e, the crossover rate, r_{co}, and the mutation rate, r_m, are set to be 0.1, 0.7, and 0.05, which are typical in the GA experiments reported in the literature [7,9]. Table 1 summarizes the values of parameters and inputs of Algorithm 1 in our GA experiments.

[5] As the Hall of Fame is empty at the beginning of each GA experiment, the first CAT game includes four individuals from the population, so the total number of games to evaluate the 20 generations is actually 199. But the difference of one game can be negligible. In theory, it is possible to design the experiments to run exactly the same number of CAT games as long as $\texttt{NUM\_OF\_GENERATIONS} \times \texttt{SIZE\_OF\_POPULATION} = 402$ and $\texttt{SIZE\_OF\_POPULATION}\%2 = 0$, however the integer solutions—201 and 2, or 67 and 6—to this equation are not practical for the GA as `SIZE_OF_POPULATION` is too small.

Table 1: The values of parameters and inputs of the GA experiments.

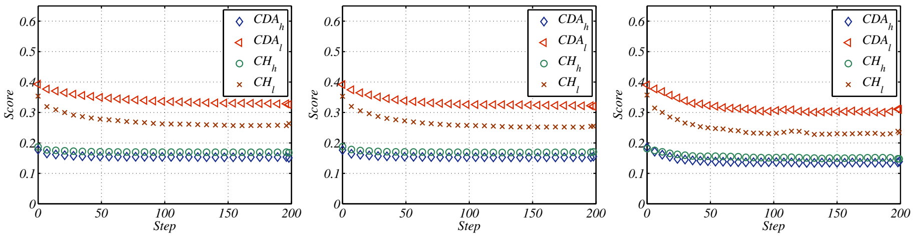
Parameter	Value	Parameter	Value
NUM_OF_GENERATIONS	20	r_e	0.1
SIZE_OF_POPULATION	20	r_{co}	0.7
NUM_OF_SAMPLES	2	r_m	0.05
NUM_OF_HOF_SAMPLES	2	$\tau^{\dagger}$	0.3
CAPACITY_OF_HOF	10	$\alpha^{\dagger}$	1
NUM_OF_POLICYTYPES	5	$\mathbb{FM}$	$\{$CH$_l$, CH$_h$, CDA$_l$, CDA$_h\}$

† τ and α are parameters in the softmax solver used by the Select($\mathbb{HOF}$, NUM_OF_HOF_SAMPLES) function, which is exactly the same in the grey-box search in [13].

To provide a better comparison, we ran two sets of GA experiments, one without crossover and the other with it. Fig. 4 and 5 show the daily scores of the four fixed auction mechanisms and the top Hall of Famers over time in two sets of GA experiments together with those from the grey-box experiments. All the results are averaged over 40 runs. Note that the x axes in the subfigures are *step* (as in the grey-box experiments), or equivalently the number of games that have been run, rather than *generation* that is common in plotting results from GA experiments. This presentation makes it easier to compare the results of the GA experiments with those from the grey-box experiments.

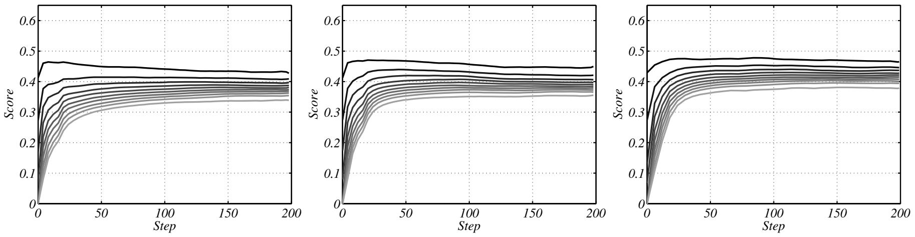
Plots in Fig. 4a and 4b, from the two sets of GA experiments respectively, exhibit the similar pattern as those in Fig. 4c, which are from the grey-box experiments. The scores of the four fixed auction mechanisms are at approximately the same positions across the three cases and then all descend until they settle down around certain values. These auction mechanisms ended up with the same relative ranking positions in these different cases. The difference is that in the end each of the four auction mechanisms settles down with different scores in different cases, the highest in the GA without crossover and the lowest in the grey-box search. This suggests that the auction mechanisms explored in the grey-box experiments are overall the most competitive while those explored in the GA experiments without crossover are the least competitive. This further indicates that the grey-box search is more effective than both versions of the GA search and as expected crossover plays an important role in the GA. Fig. 5 indicates exactly the same. Fig. 5a and 5b, from the two sets of GA experiments respectively, show that the scores of the Hall of Famers increase dramatically at the beginning of the experiments and flatten out at the end around certain positions that are lower than those in Fig. 5c.

Table 2 lists respectively the average scores of the best fixed auction mechanism, and the best and worst Hall of Famers at the end of the two versions of GA experiments and the grey-box experiments. At the 95% confidence level, any two values in the second row or any two values in the third row are significantly different from each other. That is to say that the Hall of Famers produced by the grey-box experiments are significantly better than those produced by the GA experiments. The scores of the best fixed auction mechanism in the three cases agree to this finding, but they are not significantly different. This less significance is possibly due to the fact that the CAT game is not a zero-sum game, since the transaction success rate of a mechanism in a CAT game is rel-



(a) GA without crossover. (b) GA with crossover. (c) Greybox.

Fig. 4: Scores of the four fixed auction mechanisms in the two sets of GA experiments, one without crossover and the other with crossover, and those in the first set of grey-box experiments, each averaged over 40 runs. (c) is originally Fig. 2(a) in [13].



(a) GA without crossover. (b) GA with crossover. (c) Greybox.

Fig. 5: Scores of the Hall of Famers in the two sets of GA experiments, one without crossover and the other with crossover, and those in the first set of grey-box experiments, each averaged over 40 runs. (c) is originally Fig. 2(b) in [13].

atively independent from the performance of its opponents, which counts for one third of its total score. Thus the gain of a stronger auction mechanism does not necessarily mean the same amount of loss of the losing mechanism given that all the rest of the configuration remains the same.[6]

To further investigate the effectiveness of the grey-box search in comparison with the GA search, we ran additional experiments to let the Hall of Famers produced by the grey-box experiments and the two sets of GA experiments compete against each other directly. Each of the three sets of experiments produced dozens of the Hall of Famers (69 from the grey-box experiments, 45 from the GA experiments without crossover, and 71 from the GA experiments with crossover).[7] We ran 100 CAT games with eight auction mechanisms in each game, which includes two of the fixed auction mechanisms, CDA$_l$ and CH$_l$, and two randomly selected auction mechanisms from each of the three set of Hall of Famers. Other than this, the CAT games are configured exactly the same as

[6] One example is that the scores of CDA$_h$ and CH$_h$ flatten out much earlier during the experiments than the scores of CDA$_l$ and CH$_l$ in all the three cases in Fig. 4.

[7] A Hall of Famer may come from more than one run of the same experiment.

Table 2: The average daily scores of the best fixed auction mechanism and the best Hall of Famers in the CAT games at the end of the GA experiments, and those at the end of the first set of grey-box experiments. In parentheses are the standard deviations. The scores in the second row are significantly different from each other at the 95% confidence level and so are those in the third row.

Auction mechanism	**GA without crossover**	**GA with crossover**	**greybox**[†]
Best fixed mechanism (CDA_l)	0.3260 (0.0224)	0.3203 (0.0230)	0.3101 (0.0659)
Best Hall of Famers	0.4275 (0.0233)	0.4496 (0.0340)	0.4652 (0.0210)
Worst Hall of Famers	0.3389 (0.0255)	0.3554 (0.0192)	0.3790 (0.0219)

[†] The values in this column are originally from [13].

Table 3: The average daily scores of the Hall of Famers produced by the GA experiments and the first set of grey-box experiments in direct competition in CAT games. In parentheses are the standard deviations. The scores are significantly different from each other at the 95% confidence level.

GA without crossover	**GA with crossover**	**greybox**
0.3481 (0.0201)	0.3643 (0.0188)	0.4155 (0.0291)

we did in the grey-box experiments and the GA experiments. Table 3 lists the average daily scores of the three set of auction mechanisms. At the 95% confidence level, the scores of the Hall of Famers from the grey-box experiments are significantly higher than those from either set of the GA experiments. We showed in [13] that the grey-box search was able to find mechanisms that are stronger than well known double auction mechanisms when competing directly in CAT games and are better than mechanisms that were reported in the literature in term of various economic properties and confirmed that the grey-box search can consistently produce similar results when, for example, the capacity of the Hall of Fame varies. This work provides one more piece of evidence for the superiority of the grey-box approach by comparing the results of the grey-box experiments and those of experiments based on different versions of the classic GA.[8]

5 Discussions, Future Work, and Summary

In this section, we draw contrasts and make connections between the grey-box approach and evolutionary computational approaches including GAs, GPs, and their variants. First, we can compare our grey-box approach to prior work on automated auction mechanism and trading strategy acquisition based on simple GAs, including Cliff *et al.* [4] and Phelps *et al.* [18] as well as ours reported here. A simple GA, or SGA, evolves genomes,

[8] We actually ran additional sets of GA experiments with crossover, each with a different crossover rate, 0.1, 0.4, or 1.0, in contrast to 0.7 that was used in the GA experiments described in the text. It turned out that the GA experiments using 0.7 produced the best results and hence only the results of this set of experiments were included in the text in the comparison against those of the grey-box experiments.

or binary strings, using selection, crossover, and mutation operators, while the grey-box approach evolves a vector of quality scores, each for a pre-defined building block, and explores the solution space by biasing those building blocks that lead to better solutions. A SGA maintains a set of sampling points in the solution space and tries to arrive at points of higher fitness that are accessible by applying the operators, while the grey-box approach tends to view the solution space along multiple dimensions simultaneously, maintain a hyperplane that divides the solution space into slices, adjust the sizes of the slices, and identify and explore more in those of high fitness.

A popular theory that intends to explain the effectiveness of SGAs in many optimization domains is the *building block hypothesis*, or BBH [10,12]. The BBH argues that certain building blocks of low *order* and low *defining length*, called *schemata*,[9] in the genome play a substantial role in constructing genomes of high fitness. The operators of SGAs enable the process to concentrate sampling in subspaces that are identified by these schemata and further in the common areas of these subspaces that have increasing fitness through mixing different schemata. Based on this argument, Thierens and Goldberg [23] indicated that computational expense grows exponentially with the difficulty of the problem, in terms of the number of schemata and the orders of schemata. Efforts have been made to address this issue with SGAs and improvements to SGAs were proposed by either explicitly exploring to identify schemata or implicitly using special operators to avoid breaking possible schemata in the sampled solutions [2,11]. The grey-box approach has similarities to these advanced GAs[10] since the grey-box approach explicitly considers the building blocks for auction mechanisms and biases its search towards the corners in the search space that correspond to high quality blocks.

The idea of the grey-box approach is in particular similar to that of the *compact GA*, or CGA, which was introduced by Harik *et al.* [8]. A CGA represents the population as a probability vector, rather than as a set of binary strings, where the *i*th component of the probability vector gives the probability that the *i*th bit of an individual's genome is 1. Compared with SGAs, CGAs have compact representations and work well in practice.

The tree-based model of auction mechanisms in our work bears similarities on the surface to the tree structures in GP, though the tree structure in the former case represents the whole search space and quality scores of building blocks reflect the fitness landscape of the space while tree structures in the latter each represent one individual in the search space and contains no information themselves about how fit they are.

Another topic in evolutionary computation that is related to grey-box search is the problem of early convergence to suboptimal solutions. In the grey-box experiments, parameters of the softmax exploration method in the *n*-armed bandit problem solvers were carefully set up so that sampling in the search space starts with near randomness and gradually biases modestly towards areas that are fitter than others. Techniques employed in evolutionary computation to address the problem of premature convergence, including *fitness sharing*, *crowding*, and mutation with high rate, are based on similar considerations [10,21]. For example, fitness sharing lowers the fitness of an individual

[9] A schemata is typically represented in the form, for example, `****01*1***`, where `*` can match `0` or `1`. The defining length of a schemata is the maximal distance between bits with deterministic values, and the order of a schemata is the number of bits with deterministic values.

[10] These are sometimes called *competent genetic algorithms*.

by a certain amount, which basically reflects the number of similar individuals in the population, so that similar individuals with high fitness will not be able to prevail in the next generation. In so doing, the whole population could remain diverse, approaching multiple optima in the space in parallel if applicable. In fact, as a piece of future work, these techniques can be incorporated into our GA experiments to see if the experiments can produce similar or even better results than the grey-box experiments.

Finally, our grey-box approach should be distinguished from Ronald A. Fisher's work in *population genetics* [1]. Fisher, in his research on Mendelian inheritance, assumed that—as paraphrased by Sewall Wright[11]—

> ... each gene is assigned a constant value, measuring its contribution to the character of the individual (here fitness) in such a way that the sum of the contributions of all genes will equal as closely as possible the actual measures of the character in the individuals of the population.

Wright disagreed with the view of the linear additive contribution of genes and insisted that, based on his experimental work, genes favorable in one combination are extremely likely to be unfavorable in another. Our grey-box approach is not based on Fisher's argument, although the vector of quality scores undoubtedly converges and better auction policies would obtain higher scores if the argument holds in the case of auction mechanisms. When the argument does not hold, which we believe is the case based on our experience with the experiments using CAT games, our grey-box approach may help to obtain insights on which auction policies can make better or bad combinations, and on how to design new, better policies that work better with others.

To summarize, the main contribution of this work is that we apply two different search methods in the same solution space and make a fair comparison between the two approaches, the first piece as we are aware of in the context of experimental auction mechanism design. As the search methods are domain independent, considerations in designing our algorithms and experiments and the discussions above can shed light on the design and evaluation of similar search solutions to other domain problems.[12]

References

1. K. M. Burjorjee. The fundamental problem with the building block hypothesis. *CoRR*, abs/0810.3356, 2008.
2. Y.-P. Chen and D. E. Goldberg. Convergence time for the linkage learning genetic algorithm. *Evolutionary Computation*, 13(3):279–302, 2005.
3. D. Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, HP Research Laboratories, Bristol, England, Dec. 2001.
4. D. Cliff. Explorations in evolutionary design of online auction market mechanisms. Technical Report HPL-2003-80, HP Research Laboratories, Bristol, England, July 2003.

[11] A co-founder of the field with Fisher and a critic of Fisher's approach.

[12] Similar comparisons between reinforcement learning-based approaches and evolutionary computational approaches do exist in other domains. Further discussions of these comparisons are however beyond the scope of this paper and are not possible due to the space constraint.

5. V. Conitzer and T. Sandholm. Automated mechanism design for a self-interested designer. In *Proceedings of the Fourth ACM Conference on Electronic commerce (EC'03)*, pages 232–233, New York, NY, USA, 2003. ACM.
6. S. Forrest. Genetic algorithms: Principles of adaptation applied to computation. *Science*, 261:872–878, August 1993.
7. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
8. G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3:523–528, 1999.
9. R. L. Haupt and S. E. Haupt. *Practical genetic algorithms*. John Wiley and Sons, Inc., second edition, 2004.
10. J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, USA; reprinted by MIT Press, Cambridge, MA, 1992, 1975.
11. Z. Li and E. D. Goodman. Exploring building blocks through crossover. In L. Kang, Z. Cai, X. Yan, and Y. Liu, editors, *ISICA*, volume 5370 of *Lecture Notes in Computer Science*, pages 707–714. Springer, 2008.
12. M. Mitchell, S. Forrest, and J. H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In F. V. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: First European Conference on Artificial Life*, pages 245–254. MIT Press, Cambridge, MA, 1991.
13. J. Niu, K. Cai, and S. Parsons. A grey-box approach to automated mechanism design. In *Proceedings of the Twelfth Workshop on Agent-Mediated Electronic Commerce (AMEC XII)*, Toronto, Canada, 2010.
14. J. Niu, K. Cai, S. Parsons, E. Gerding, and P. McBurney. Characterizing effective auction mechanisms: Insights from the 2007 TAC Mechanism Design Competition. In Padgham et al. [17], pages 1079–1086.
15. J. Niu, K. Cai, S. Parsons, E. Gerding, P. McBurney, T. Moyaux, S. Phelps, and D. Shield. JCAT: A platform for the TAC Market Design Competition. In Padgham et al. [17], pages 1649–1650. Demo Paper.
16. J. Niu, K. Cai, S. Parsons, P. McBurney, and E. Gerding. What the 2007 TAC Market Design Game tells us about effective auction mechanisms. *Journal of Autonomous Agents and Multiagent Systems*, 21(2):172–203, 2010.
17. L. Padgham, D. C. Parkes, J. P. Müller, and S. Parsons, editors. *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008*. IFAAMAS, 2008.
18. S. Phelps, M. Marcinkiewicz, S. Parsons, and P. McBurney. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 705–712, New York, NY, USA, 2006. ACM Press.
19. S. Phelps, S. Parsons, E. Sklar, and P. McBurney. Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the Workshop on Agents for Electronic Commerce*, Pittsburgh, PA, 2003.
20. C. D. Rosin. *Coevolutionary Search Among Adversaries*. PhD thesis, University of California, San Diego, San Diego, CA, USA, 1997.
21. B. Sareni and L. Krähenbühl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, September 1998.
22. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
23. D. Thierens and D. E. Goldberg. Mixing in genetic algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 38–47, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.