# Fair Cake-Cutting Algorithms with Real Land-Value Data

Itay Shtechman, Rica Gonen and Erel Segal-HaLevi

**Abstract**

Fair division of land is an important practical problem that is commonly handled either by hiring assessors or by selling and dividing the proceeds. A third way to divide land fairly is via algorithms for *fair cake-cutting*. Such un-intermediated methods are not only cheaper than an assessor but also theoretically fairer since they guarantee each agent a fair share according to his/her value function. However, the current theory of fair cake-cutting is not yet ready to optimally share a plot of land, and such algorithms are seldom used in practical land division.

We attempt to narrow the gap between theory and practice by presenting several heuristic adaptations of famous algorithms for one-dimensional cake-cutting to two-dimensional land-division. The heuristics are evaluated using extensive simulations on real land-value data maps from three different data sets and a fourth (control) map of random values. The simulations compare the performance of cake-cutting algorithms to sale and assessor division in various performance metrics, such as utilitarian welfare, egalitarian welfare, envy, and geometric shape. The cake-cutting algorithms perform better in most metrics. However, their performance is greatly influenced by technical implementation details and heuristics that are often overlooked by theorists.

We also propose a new protocol for practical cake-cutting using a dynamic programming approach and discuss its runtime complexity versus performance trade-off. Our new protocol performs better than the examined classic cake-cutting algorithms on most metrics. Experiments to assess the amount that a strategic agent can gain from reporting false preferences are also presented. The results show that the problem of strategic manipulation is much less severe than the worst-case predicted by theory.

## 1 Introduction

Fair division of land is an important practical problem. It occurs in inheritance cases, partnership resolutions, and public land allocations. It is commonly handled by hiring the services of expert human assessors, who calculate a division that is considered fair based on elaborate land-value assessments. If the partners do not want to trust an assessor, they often sell the land in the market and split the revenues.

But there is a different way to divide land fairly. In the last 70 years, economists and computer scientists have developed various algorithms for fair *cake-cutting* - fair division of a continuous heterogeneous resource among agents with different preferences **(Section 3)**. Using such automatic methods is not only cheaper than renting a human assessor — it is also theoretically fairer, since it guarantees to each agent a fair share by his/her personal value function, which may differ from the assessor's value function.

As far as we know, these theoretically-superior algorithms are seldom used in practical land-division problems. Our goal is to bring theory and practice closer together, and this paper describes several steps we made in this direction.

First, we constructed instances of cake-cutting problems based on real land-value data. We constructed detailed two-dimensional land-value maps of New Zealand, most of Israel and a higher resolution map of Tel-Aviv city.

To simulate agents with different but correlated valuations, we constructed, from each

land-value map, $n$ different maps (one map per agent), where each map is based on the original map and some random noise. We experimented with various values of $n$ between 4 and 128, two noise models, and various noise levels, ranging between 20% and 60% of the base value **(Section 4)**.

Second, we adapted the classic cake-cutting algorithms of Even and Paz (1984) and Steinhaus (1948) to divide a two-dimensional map into rectangular land-plots. This adaptation can be done in many ways, since each cut of the interval made by the original algorithm can be converted to a horizontal (east-west) cut or to a vertical (north-south) cut of the two-dimensional map. The cut direction can be fixed in advance, or it can be decided dynamically based on various heuristics such as cutting the longer side or choosing the cut that maximizes the social welfare. Most previous works that we know of ignored these questions since they assumed that the cake is one-dimensional. However, when dividing a two-dimensional land, such decisions may affect the allocation quality.Therefore, we present a dynamic programming algorithm for finding the sequence of $n-1$ cut-directions that optimizes a given performance metric **(Section 5)**. The run-time of the algorithm is $O(n^2)$, which may be slow for large values of $n$. Therefore, we implemented over 10 different heuristics for choosing the cut-direction, which allow the algorithm to run in time $O(n \log n)$.

Third, we conducted extensive experiments with different numbers of agents and four different maps (the above three and a randomly-generated map). In each setting, we compared the cake-cutting algorithm in all its different variants to the two baseline methods commonly used today for dividing land: (1) *assessor division*, where the land is partitioned into pieces with the same base value (ignoring the "noise") and each agent receives one piece; (2) *market sale*, where the land is sold for its total market value and each agent receives $1/n$ of the proceeds. We compared the methods using several metrics: *utilitarian welfare*, *egalitarian welfare*, and *envy*.

We found substantial differences in performance among the different heuristics. For each performance metric, one or two heuristics were clearly superior to the others. This shows the importance of modelling land in two dimensions rather than reducing it to a 1-dimensional interval. The cake-cutting algorithms fared similar or better than assessor division in all metrics, and better than market sale in the two welfare metrics **(Section 6)**.

A potential problem with cake-cutting algorithms, that does not exist in market sale or assessor division, is that the agents may try to manipulate the algorithm by misrepresenting their preferences. We measured the *strategic gain* — the amount by which a strategic agent, who knows the actions of all other agents in the current round and plays a best response, can increase his/her utility relatively to a truthful agent. The average strategic gain of such an agent was less than 2.2% (same order of magnitude as the payment to a real-estate broker). In the more realistic situation, in which an agent does not know the actions of all other agents, strategic manipulations would probably be even less profitable, and may even result in losing value. This information is helpful, as it allows us to advise participants in cake-cutting algorithms to report their valuations truthfully even though theoretically the algorithm is not truthful.

The advantage of the cake-cutting algorithms was much more pronounced in the two maps based on land values than in the random map. This illustrates the importance of using real value data for evaluating fair division algorithms. This is in contrast to previous work on simulation of fair division algorithms, which mostly used artificially-generated data.

The improvements provided by the cake-cutting algorithm are encouraging. They imply that the algorithmic "do it yourself" approach to fair division, which is cheaper than the common approaches of selling or employing an expert assessor, is also better in terms of social welfare and reducing envy. Still, we believe that there is a lot of room for future improvement by using sophisticated cake-cutting algorithms that are specifically tailored to perform well in a specific metric (**Section 7**).

# 2 Related Work

Surveys on the theory of cake-cutting algorithms can be found in Brams and Taylor (1996); Robertson and Webb (1998); Brams (2007); Brânzei (2015); Lindner and Rothe (2016) and Procaccia (2016). Below we focus on the more practice-oriented works on fair division. These can be categorized into computerized simulations, case-studies and lab experiments.

## 2.1 Computerized Simulations

Fair division papers often use simulations to compare the performance of various algorithms. For example, Cavallo (2012) used simulations to compare his redistribution-based improvement of VCG to the original VCG w.r.t. fairness and welfare. Dickerson et al. (2014); Aziz et al. (2020) used simulations to check under what conditions a fair assignment of discrete objects exist with high probability.

In these works, the agents' valuations were drawn at random. Correlation between different valuations for the same item was modeled by assigning to each item a random *intrinsic value*, and adding a random noise for each agent.

Our work is similar to these works in that it, too, compares different methods for fair division using computerized simulations. It differs in that, in our model, the "intrinsic value" of each land-plot is not selected at random but rather based on real land values.

## 2.2 Case Studies

Flood (1958) studied a case of dividing gift parcels using the Knaster algorithm (Steinhaus, 1948).

Several counter-factual studies checked the feasibility of using the Adjusted Winner fair-division protocol for resolving international disputes: Camp David Accords, Spratly Islands controversy and the Israeli-Palestinian conflict (Brams and Taylor, 1996; Brams and Togman, 1996; Brams and Denoon, 1997; Massoud, 2000).

Budish and Cantillon (2007, 2012) studied the course allocation mechanism in the Harvard business school. They found out that the mechanism is often manipulated, and this leads to Pareto-inefficiency and a significant loss of welfare.

Gal et al. (2017) test their algorithm for fair division of rooms and rent on data collected from users of the spliddit.org website (Goldman and Procaccia, 2015). Oluwasuji et al. (2018) test their heuristic algorithms for *fair load-shedding* on electricity-usage data, which they collected from a USA-based database and adapted to African consumption patterns.

Many authors test their algorithms on the PrefLib library (Mattei and Walsh, 2013) — a collection of real-world preference relations on discrete items.

All these works consider the allocation of discrete items (possibly with money). We contribute to this literature by studying the allocation of land, which is a *continuous good* that requires quite different division algorithms.

## 2.3 Laboratory Experiments

In many experiments, agents are asked to divide items in the lab using several pre-specified procedures. The experimenters then compare different division procedures according to various criteria, such as: perceived fairness, economic efficiency, etc. (Schneider and Krämer, 2004; Dupuis-Roy and Gosselin, 2011). Other experiments test what desiderata are more important to the subjects when there is a conflict, e.g. between fairness and efficiency (Engelmann and Strobel, 2004; Fehr et al., 2006; Herreiner and Puppe, 2010). Some experiments check whether participants choose to play strategically, and how their manipulations

affect the fairness of the final outcome (Parco and Rapoport, 2004; Daniel and Parco, 2005; Hortala-Vallve and Llorente-Saguer, 2010; Kyropoulou et al., 2019).

In lab experiments, due to practical reasons, the division instances are small (e.g. Kyropoulou et al. (2019) report a cake-cutting experiment with at most 4 participants per instance). In contrast, in our simulation experiments we could compare division procedures on much larger instances.

# 3 Preliminaries

A land-estate $C$ ("cake") has to be divided among $n$ agents. Each agent $i$ has a value-density function $v_i$, mapping each point of $C$ to its monetary value for $i$. We define for each agent $i$ its value-measure $V_i$, which maps each *subset* of $C$ to its monetary value for $i$. The value-measure is the integral of the value-density, i.e., for each subset $X \subseteq C$, $V_i(X) = \int_{x \in X} v_i(x)dx$. Hence, each value-measure is an additive set function — the value of a union of two disjoint plots is the sum of the values of the plots. Additivity is crucial to the operation of most cake-cutting algorithms.

We assume that the agents' valuations are *piecewise-constant*: $C$ is partitioned into small rectangular cells, and in each cell, the value-density of each agent is constant. Thus $V_i$ can be represented by a matrix of the cell values, where $V_i$ of a union of cells is the sum of the cells' values.

$C$ should be partitioned into $n$ disjoint contiguous pieces, $X_1, \ldots, X_n$, one piece per agent. $C$ is assumed to be a rectangle, and each piece should be a rectangle too.

## 3.1 Evaluation Metrics

There are infinitely many land partitions, and there are various metrics by which a partition can be evaluated. Our metrics use the valuation of each agent relative to the agent's valuation of $C$.

The **Utilitarian Value** of a partition $X$ is the sum of the agents' relative values:

$$\mathrm{UV}(X) = \sum_{i=1}^{n} V_i(X_i)/V_i(C)$$

The **Egalitarian Value** of $X$ is the smallest relative value of an agent (multiplied by $n$ to make it comparable to $UV$):

$$\mathrm{EV}(X) = n \min_{1 \le i \le n} V_i(X_i)/V_i(C)$$

A partition $X$ is called *proportional* if $EV(X) \geq 1$.

For every two agents $i$ and $j$, we calculate the *Envy* of agent $i$ in agent $j$ by comparing the value agent $i$ got from his/her own land-plot to the value he/she could get from having agent $j$ 's plot:

$$\textsc{Envy}(i,j) = \begin{cases} V_i(X_j)/V_i(X_i), & \text{if } V_i(X_j) > V_i(X_i) \\ 1 & , \quad \text{Otherwise} \end{cases}$$

The **Largest Envy** of a partition $X$ is defined as:

$$\mathrm{LE}(X) = \max_{1 \le i \le n} \max_{1 \le j \le n} \textsc{Envy}(i,j)$$

As a baseline for comparison, note that if the land is sold and the money is divided among the agents, the relative value of each agent is exactly $1/n$, so $UV = EV = LE = 1$ (we defined the metrics such that their baseline is 1).

# 4 Maps

We constructed four land-value maps.

The first map was of New Zealand. We used the Forest Profit Expectations Dataset (U10073) published by Motu — a New Zealand institute for Economic and Public Policy Research (http://motu.nz). It contains 3000-by-2000 matrix where each element represents a 500-by-500 meters land-block. The value describes the Net Present Value of planting a forest in that land-block. The total matrix corresponds to a 1000-by-1500 kilometers rectangle which entirely contains New Zealand. We assume free disposal so blocks in the sea, as well as land blocks with negative NPV, were set to 0. In our experiment we reduced the dataset resolution due to cache memory limitations on our computational machines, so we summed each four adjacent matrix elements to create a single element representing a 1000-by-1000 meters land-block. All in all, we got a 1500-by-1000 matrix of non-negative values representing the estimated profit for an entrepreneur planting a forest in each spot.

The second map was of Israel. We found a database constructed by Madlan —a commercial Israeli website for classified real-estate ads (http://madlan.co.il). It contains a list of all major neighborhoods in Israel and their estimated PPM (price per square meter) calculated from a database of past apartment sales. We extracted each neighborhood latitude and longitude using a map of Israel. We created a 1150-by-1000 matrix representing the land of Israel —each element in the matrix corresponds to a 360-by-160 meters land-block that was initialized with a PPM based rough estimate of land value.

Arguably, it is not very realistic that an entire country will be divided. For that reason, the third map we constructed was of Tel-Aviv city. We used the same list of estimated neighborhoods' PPM and Tel-Aviv's city plans to construct a map where each neighborhood area received the appropriate PPM value resulting in a higher resolution real estate map of a relatively small area (52 squared kilometers).

For all three land-maps described, the values do not represent the actual market-price of each land-block. In New Zealand, many land-blocks already have planted forests, buildings or other improvements. In Israel and Tel-Aviv, our calculations did not take into account the neighborhood surrounding aspect. However, the land values in our map still show a *dependence on geographic location* and maintain interesting properties. For comparison, we generated a 1500-by-1000 land by choosing a value for each matrix element uniformly at random between 0 and 1,000,000.

Fig. 1 shows heat-maps illustrating the four maps. It shows that, while the random map is uniform, the rest are far from uniform: they have small regions with a high value and large regions with a small value.

Land-value maps represent an "objective" evaluation of the land. But, people's preferences have a "subjective" component too. We capture this subjective component by a random noise that is added to the land-value map. To control the relative weight of the subjective versus the objective component, we introduce the noise-ratio parameter, denoted by $r$. From each land-value map we created two different datasets, each dataset containing $n$ variants, one variant per agent. Each one of the two datasets was created using one of two noise patterns:

**Uniform**: For each variant, we selected the value for each land-block uniformly at random between $(1-r)V$ and $(1+r)V$ , where $V$ is taken from the land-value map. This noise pattern represents agents with different "subjective" valuations but no clear specific area of interest.

**Hotspot**: For each variant, a map land-block coordinate $(x_c, y_c)$ was selected uniformly at random. The goal was to represent a subjective preference for an agent specific area of interest. The point $(x_c, y_c)$ represents a spot that the agent finds particularly valuable; the agent's valuation of other points is larger the closer they are to this "hot spot". The noise

distance is limited using $r$. The formal definition of the "hot spot" noise function is given in the appendix.

Fig. 2 show heat-maps illustrating the hot spot noise pattern for different values of noise parameter $r$.

All in all, for each of the four base maps, we can evaluate the algorithm performance over either a dataset created using hotspot noise pattern or a dataset created using uniform noise pattern. Each dataset is made of $n$ matrices of non-negative values, each representing a different piecewise-constant value measure.

# 5 Algorithms and Heuristics

## 5.1 Basic Algorithms

There are many algorithms for finding a proportional cake-partition. In this experiment we focus on the *Even-Paz (EP)* algorithm (Even and Paz, 1984), since we believe it is the algorithm most likely to be used in a cake-cutting problem with a large number of agents. This is due both to its simplicity and to its optimal run-time complexity: it runs in time $O(n \log n)$, which is provably the best possible (Woeginger and Sgall, 2007). For simplicity we present the algorithm for an even $n$ (in our experiments $n$ was always a power of 2). At the first iteration, each agent $i$ is asked to make a *query-mark* $x_i$ — a mark that divides $C$ into two pieces with equal value in his/her eyes. The algorithm orders the marks and renumbers them such that $x_1 \leq \ldots \leq x_n$, then calculates the median $x_M := (x_{n/2} + x_{n/2+1})/2$. The algorithm cuts $C$ at $x_M$, producing two sub-cakes, and recursively divides each sub-cake among the $n/2$ agents whose query-mark is in that part. When $n = 1$, the one agent receives all $C$.

We compare the results of Even-Paz to another classic proportional cake-cutting algorithm — *Last-Diminisher (LD)* (Steinhaus, 1948).Similar to Even-Paz, at the first iteration, each agent $i$ is asked to make a *query-mark* $x_i$, only that the mark is made to divide $C$ into two pieces with value ratio $1 : n - 1$ in his/her eyes. The algorithm orders the marks and renumbers them such that $x_1 \leq \ldots \leq x_n$, then cuts $C$ at $x_1$, producing a piece for the agent that proposed query-mark $x_1$ and a sub-cake which is recursively divided among all other $n - 1$ agents.

## 5.2 Adaptations to a 2-Dimensional Setting

Both Even-Paz and Last-Diminisher are well-defined for a one-dimensional cake. However, when $C$ is two dimensional, in each iteration, the agents can make their query-marks in many different directions; the only requirement is that the query-marks of all agents at a given iteration do not intersect (so that they can be ordered). Even with the natural restriction that the cuts should be parallel to the axes (i.e., either horizontal or vertical), there are still $2^{n-1}$ options — two options per cut (for example, with $n = 4$ agents, there are two options for the first cut. This cut yields two sub-cakes, and in each of these sub-cakes there are two options. All in all there are $2^3$ options). How can a practitioner decide which of these $2^{n-1}$ sequences to use?

One option is to decide on a certain metric to optimize, such as the utilitarian value. Given a specific instance, one can try all $2^{n-1}$ sequences and choose the one that yields the highest utilitarian value. We did this exhaustive search for $n = 4$ and $n = 8$ agents. However, for larger values of $n$ this becomes impractical.[1]

[1]Note that even this exhaustive search does not yield the maximum possible utilitarian value overall. In fact, even with a 1-dimensional cake, neither Even-Paz nor Last-Diminisher yield the maximum utilitarian value; finding an allocation that maximizes the utilitarian value is NP-hard. The same is true for egalitarian

To avoid such exhaustive search we propose the use of a dynamic programming approach in order to find such optimal cut sequence. For simplicity we present the *FindOptimalCut-Sequence (FOCS)* algorithm for an $n$ which is a power of 2. The formal definition of FOCS and proof that it is optimal is given in the appendix.

## 5.3 Cut-direction heuristics

We tested two sets of heuristics: predefined heuristics and greedy heuristics. The predefined heuristics are:

**Hor**: all $n-1$ cuts are horizontal (east to west).

**Ver**: all $n-1$ cuts are vertical (north to south).

**HorVer**: The first cut is horizontal; in each recursion level the direction switches.

**VerHor**: As above but the first cut is vertical. HorVer and VerHor heuristics were chosen in hope that they would have a better face-ratio than Hor and Ver.

For the *greedy heuristics*, each agent $i$ makes two query-marks in each iteration: a vertical query-mark $x_i$ and a horizontal query-mark $y_i$. We define the *proposed vertical cut* $x_P$ and the *proposed horizontal cut* $y_P$ according to the cut location selection of the examined algorithm. When using Even-Paz, $x_P := x_M$ and $y_P := y_M$. When using Last-Diminisher $x_P := x_1$ and $y_P := y_1$. We then select the cut by one of the following heuristics.

**SmallestCut**: Cut vertically at $x_P$ if $x_P < y_P$; else cut horizontally at $y_P$.

**SmallestPiece**: Make a cut resulting in a smaller piece. Let $x_A$ be the area to the left of $x_P$ and $y_A$ the area above $y_P$. Cut vertically at $x_P$ if $x_A < y_A$; else cut at $y_P$.

**HighestScatter**: Define the vertical scatter as $x_S := x_n - x_1$ and the horizontal scatter as $y_S := y_n - y_1$. Cut vertically at $x_P$ if $x_S > y_S$; else cut horizontally at $y_P$.

**SquarePiece**: Choose a cut direction resulting in a land-plot with a larger face ratio. Let $x_F$ be the average of the face-ratios of the piece to the left and the piece to the right of $x_P$. Let $y_F$ be the average of the face-ratios of the piece above and the piece below $y_P$. Cut vertically at $x_P$ if $x_F > y_F$; else cut horizontally at $y_P$.

Two additional greedy heuristics we examined are based on the notion that local benefit of a specific iteration is influenced by the margin between the two query-cuts from both sides of the proposed cut. When using Even-Paz the vertical margin is the area between $x_{n/2}$ and $x_{n/2+1}$, and the horizontal margin is the area between $y_{n/2}$ and $y_{n/2+1}$. For Last-Diminisher the vertical margin is the area between $x_1$ and $x_2$, and the horizontal margin is the area between $y_1$ and $y_2$.

**LargestMargin**: Choose a cut direction resulting in a larger margin. Cut vertically at $x_P$ if the length of the vertical margin is greater than the length of the horizontal margin; else cut horizontally at $y_P$.

**MostValuableMargin**: Choose a cut direction resulting in a more valuable margin. Cut vertically at $x_P$ if the average value of the vertical margin (averaged over all $n$ agents) is larger than the average value of the horizontal margin; else cut horizontally at $y_P$.

**Post-processing heuristic.** For each examined algorithm, after the allocation is found, we run a post-processing step where we allow for mutually-beneficial exchanges of allocated pieces to take place. The exchange is handled using the *Top Trading Cycle (TTC)* algorithm (Shapley and Scarf, 1974).

value (Aumann et al., 2013). However, since the focus of the present work is the adaptation of one-dimensional algorithms to a two-dimensional cake, we consider the maximum utilitarian value attained by one of the $2^{n-1}$ possible cut-direction sequences as "optimal".

Table 1: MostValuableMargin UV comparison to assessor, market-sale and FOCS, n=128

| | New Zealand | | Israel | | Random | |
|---|---|---|---|---|---|---|
| | Uni. | HS | Uni. | HS | Uni. | HS |
| Assessor comparison | 2.8% | 23.9% | 2.6% | 26.6% | 0.8% | 56.6% |
| Market comparison | 2.9% | 24% | 2.6% | 26.6% | 0.8% | 57.1% |
| FOCS comparison | -0.1% | -0.2% | -0.1% | -0.4% | -0.02% | -0.6% |

Table 2: MostValuableMargin EV comparison to assessor, market-sale and FOCS, n=128

| | New Zealand | | Israel | | Random | |
|---|---|---|---|---|---|---|
| | Uni. | HS | Uni. | HS | Uni. | HS |
| Assessor comparison | 4.7% | 71.3% | 4.5% | 66.3% | 1.25% | 48.9% |
| Market comparison | 0.9% | 6.4% | 0.9% | 6.3% | 0.3% | 28.7% |
| FOCS comparison | -0.6% | -1.3% | -0.4% | -3.8% | -0.2% | -4.5% |

**Assessor division.** We compare our algorithms' results to the performance of an assessor. To simulate an assessor division, the land is divided into $n$ parts with the same base-land values, ignoring the random noise. The division is done using the following simple method. For some integers $k_1, k_2$ with $k_1 \cdot k_2 = n$, the assessor first partitions the land using vertical cuts into $k_1$ parallel strips of value $1/k_1$, and then partitions each such strip using horizontal cuts into $k_2$ plots of value $(1/k_1)/k_2 = 1/n$. We take $k_1 = 2^{\lceil (\log_2 n)/2 \rceil}$ and $k_2 = 2^{\lfloor (\log_2 n)/2 \rfloor}$, so that the pieces have a balanced aspect ratio. In particular, if $n$ is a square number then $k_1 = k_2 = \sqrt{n}$.

# 6 Experiments and Results

We ran experiments for each of the three maps, for $n$ in {4, 8, 16 ,32, 64, 128}. For each map and $n$, we ran 50 experiments with different randomly-generated noise. In most experiments the noise-ratio was $r = 0.6$. In each experiment, we ran each heuristic twice: once using Even-Paz algorithm and once using Last-Diminisher. All in all we tested 1,800 different experiment settings (3 dataset x 2 noise-patterns x 50 repetitions x 6 agent-numbers) and calculated 39,600 partitions (1,800 experiments x (2 algorithms x 10 heuristics + FOCS + Assessor)). For each partition, we calculated the evaluation metrics UV, EV, LE, AFR, SFR and Run Duration (RD). For each setting and each metric, we calculated its average over the 50 runs and a 95% confidence interval. All reported $p$-values were calculated using a two-tailed t-test.

The results for the New Zealand, Israel and Tel-Aviv maps were very similar, but the results for the random map were substantially different. Below, we present graphs illustrating the results for New Zealand and selected results of Israel and random maps as well.

## 6.1 Utilitarian and Egalitarian Value

Fig. 3 compares the UV of various heuristics. All heuristics score better than the baseline of $UV = 1$. Besides FOCS, which finds the optimal cut-sequence, the best-performing heuristic with both uniform and hotspot noise patterns is MostValuableMargin. Intuitively, the reason is that MostValuableMargin takes the most advantage of the differences between the agents' valuations.

MostValuableMargin is only slightly better than the other greedy heuristics ($p < 0.1$), but significantly better than the predefined heuristics ($p < 0.001$). The $UV$ of Even-Paz

+ MostValuableMargin is significantly better than the market-sale ($p < 0.001$) and the advantage grows with $n$.

Table 1 contains the advantage of Even-Paz + MostValuableMargin over assessor and market-sale and the disadvantage compared to FOCS for the different maps and noise patterns with $n = 128$. The UV of an assessor division is very close to 1 – similar to market-sale (see Fig. 4).

The difference in the results between the hotspot noise and the uniform noise might be explained by the nature of both patterns. For example, if the land is being divided between two agents using hotspot noise, each agent would likely have a different area of interest as his/her hotspot, making it fairly simple to divide the land in such a way that each agent will receive their respective area of interest resulting in a high UV. On the other hand, when the noise is spread out uniformly at random it is harder to choose a similarly beneficial allocation.

The results for random map using hotspot noise are more then twice the results for the other maps, this illustrates the importance of using real land-value data to evaluate division algorithms in order to avoid over-optimistic results.

The graphs of EV for the different heuristics are qualitatively similar to the graphs of UV. The best heuristic is again MostValuableMargin (see Fig. 5).

The EV of Even-Paz is always higher than the EV of selling the land and with MostValuableMargin it is significantly better ($p < 0.001$). For uniform noise, the advantage of Even-Paz grows with $n$ but with hotspot noise, the advantage does not grow as $n$ grows; this trend can be explained by further examination of the hotspot noise pattern. Recall that we have created each agent's map by selecting a random coordinate and generated a cone-shaped noise pattern around it. As a result of this pattern, coordinates on the map that are far away from the noise center have little or no noise added to it creating areas on their "subjective" maps that are equivalent to the "objective" evaluation of the land, this means that two agents can have high correlation in their preferences for a subset of the land that mainly contains "objective" evaluations for them both, if those two agents were asked to divide that subset of land between them using the Even-Paz algorithm, both will propose a similar cutting location and the land will be divided accordingly resulting in each agent receiving roughly 50% of the land-plot resulting in a small EV.

Table 2 contains the advantage of Even-Paz + MostValuableMargin over assessor and market-sale and the disadvantage compared to FOCS for the different maps and noise patterns with $n = 128$.

The EV of an assessor division is always *lower* than of selling the land. The dis-advantage grows with $n$; with $n = 128$ it is about $-37.8\%$ for hotspot noise and $-3.7\%$ for uniform noise (see Fig. 5). This means that, in an assessor division, there is always at least one person who receives less than his/her fair share. Even receiving 3.7% below the fair share might make a person feel frustrated for being treated unfairly. This cannot happen when using the Even-Paz cake-cutting algorithm, since by design it always gives each agent at least a fair share.

Applying the TTC algorithm to the outcome of Even-Paz + MostValuableMargin slightly improved the UV with uniform-noise, but had no noticeable effect on the UV with hotspot-noise or on the EV.

## 6.2 Largest Envy

Fig. 6 compares the LE of various heuristics.

Note that smaller LE is better, and market-sale always trivially attains the minimum LE which is 1. Besides market-sale, MostValuableMargin attains the smallest LE of the heuristics we tested, but the advantage is not statistically significant ($p = 0.2$).

Table 3: MostValuableMargin LE comparison to assessor and market-sale, n=128

| | New Zealand | | Israel | | Random | |
|---|---|---|---|---|---|---|
| | Uni. | HS | Uni. | HS | Uni. | HS |
| Assessor comparison | 3.5% | 38.4% | 3.35% | 34.6% | 1.0% | 29.6% |
| Market comparison | -4.1% | -20.3% | -4.7% | -27.7% | -1.0% | -30.8% |

Table 4: Strategic Gain with Even-Paz algorithm, New Zealand Map, Hotspot Noise, n=128

| Cut-direction heuristic | Average strategic gain | Maximum strategic gain | Strategy risk |
|---|---|---|---|
| Ver | 0.9% | 7.2% | 0.2% |
| Hor | 0.9% | 4.5% | 0.1% |
| VerHor | 1.9% | 7.0% | 0.4% |
| HorVer | 1.7% | 5.8% | 0.2% |
| MostValuableMargin | 1.9% | 8.2% | 4.3% |
| LargestMargin | 2.2% | 15.5% | 28.6% |
| SmallestPiece | 1.1% | 4.9% | 0.1% |
| HighestScatter | 1.8% | 6.0% | 1.5% |
| SquarePiece | 1.8% | 8.3% | 2.7% |
| SmallestCut | 0.9% | 4.6% | 0.1% |

The LE of both Even-Paz and assessor-division grows as $n$ grows, but the LE of Even-Paz is better (smaller, see Fig. 7). For $n = 128$, Even-Paz + MostValuableMargin is significantly smaller than assessor division ($p < 0.001$).

Table 3 contains the advantage of Even-Paz + MostValuableMargin over assessor and the disadvantage over market-sale for the different maps and noise patterns with $n = 128$.

Since FOCS does not necessarily find the optimal cut-sequence for LE (see appendix) , we calculated the optimal cut-sequence by exhaustive search for $n = 4$ and $n = 8$ (for $n \geq 16$ the run-time is too large). For $n = 8$, the LE of the optimal cut-sequence (above the baseline of 1) was 2.5%, as opposed to 6.3% for MostValuableMargin.

## 6.3 Comparison to Last-Diminisher

We ran the same set of experiments with the Last Diminisher algorithm instead of Even-Paz. Below we highlight the main similarities and differences.

- The UV and LE were similar to Even-Paz. For UV the best-performing heuristic was still MostValuableMargin and for LE it was LargestMargin.
- The EV was worse ($p < 0.001$). Because Last-Diminisher is designed to allocate the first agent exactly $1/n$ of $C$ in his/her eyes, its EV always equals 1.

All in all, Even-Paz is at least as good, and often better, than Last-Diminisher.

## 6.4 Strategic Gain

It is known that most existing cake-cutting algorithms are not *strategyproof*, which means that agents may gain by strategically reporting false preferences (Brams and Taylor, 1996; Menon and Larson, 2017). In theory, the gain from such manipulation can be very high: there are instances in which a truthful agent receives only $1/n$ of the total value, while a strategic agent receives 100% of the value (Kyropoulou et al., 2019). In order to get a more

realistic estimate, we ran experiments on our land-value based maps to check how much a single agent may gain by being strategic, assuming all other agents are truthful.

Calculating the strategic gain requires us to make some assumptions on how much information the agent has on the preferences of other agents, and how sophisticated the agent is calculating an optimal strategy. On one hand, a worst-case analysis would require us to assume that the strategic agent knows the value functions of all other agents, and uses a globally-optimal strategy. On the other hand, recent laboratory experiment (Kyropoulou et al., 2019) shows that humans are far from optimal in manipulating cake-cutting algorithms. As a middle ground, we decided to experiment with *locally-optimal* strategies.

Suppose the strategic agent is Alice, while the other $n-1$ agents are truthful. We assume Alice is acting strategically in a local manner, meaning at each iteration Alice makes a greedy selection of her query-mark to maximize her value. The strategy of Alice for selecting her query-marks is calculated in two steps, at the first step Alice forces a better cut location in both possible cut directions and at the second step of her strategic selection, if the cutting heuristic is not a predefined one, Alice tries forcing the cut direction to further expand her local gain. The complete description of the strategy is given in the appendix.

The **Strategic Gain** of agent $i$ is the increase in utility of agent $i$ while playing strategic. We calculate the strategic gain of agent $i$ by conducting a standard truthful experiment followed by $n$ additional strategic experiments, one for each agent $i$.

The **Average** and **Maximum Strategic Gain** over all agents are defined by:

$$\text{ASG} = \sum_{i=1}^{n}(SG_i)/n \quad ; \quad \text{MSG} = \max_{1\leq i\leq n}(SG_i)$$

Note that, since our strategic agent acts greedily, she may fail to find the globally optimal manipulation. This means that she may gain less by acting strategically than by acting truthfully. We measure the **Strategy Risk**, which is the average percentage of agents who gained *less* by being strategic as opposed to being truthful.

Table 4 shows the results for 128 agents, New Zealand map and hotspot noise. The largest MSG in all our experiments was 15.5% (in New Zealand map, hotspot noise, 128 agents, LargestMargin heuristic). However, the ASG in all experiments was less than 2.2%.

For each one of our tested heuristics, there were agents who gained less by acting strategic than by being truthful. The LargestMargin heuristic had a significantly higher strategy risk then all other heuristics (28.6% in New Zealand map, hotspot noise, 128 agents), followed by MostValuableMargin, SquarePiece and HighestScatter. This is especially interesting because those heuristics has a generally higher MSG, suggesting a correlation between risk and reward in a strategic play. We believe that the high strategy risk in some of the tested heuristics can be explained by the magnitude by which the entire run of a tested heuristic is affected by each greedy selection of the strategic agent.

Our results suggests that, in general, strategic manipulations are not a big issue for land division algorithms — even when agents have complete knowledge of all other agents' preferences, they would be required to make a big effort to slightly increase their gain (in our experiments about 2.2%) and by doing so taking a risk of gaining less than they could have. In practice agents do not have complete knowledge of others' preferences, so it is even harder to gain from strategic manipulation.

# 7 Conclusions and Future Work

Our results have several practical implications for people dealing with fair division of land.

1. Cake-cutting algorithms may be a viable alternative to the common methods for land division, namely assessor division and market-sale. In particular, Even-Paz attains

higher social welfare than both these methods, and lower envy than assessor division. The effect is larger when there are more agents and when there is more variation in the agents' valuations. Even-Paz performs better than Last Diminisher.

2. When adapting cake-cutting algorithms to two dimensions, FOCS performs well in most metrics, but has to be tailored to a specific performance measure for each different run and also, takes longer to execute than other tested methods. Of the faster heuristics, MostValuableMargin is superior in terms of social welfare and envy, while SquarePiece is superior in terms of geometric shape. This shows a trilateral trade-off between run-time, and social welfare.

3. When assessing cake-cutting algorithms using artificially generated data, results are highly effected by the methods used to generate the data. Randomly created maps can give overly optimistic or pessimistic results compared to real land-value data. Furthermore, the performance of the algorithms are also highly influenced by the chosen method for adding randomly generated noise in order to represent different agents' valuations. Using a hotspot in our case to represent *desired point of interest*, the cake-cutting algorithms yielded much better results than when using uniform random noise. In order to correctly assess how cake-cutting algorithms perform, we must model the preferences of our agents in as realistic way as possible.

4. Strategic manipulation may improve the welfare of an agent with complete information, but the improvement is minor. Greedy strategic manipulation can effect the entire run of the algorithm and result in smaller gain for the agent being strategic — the risk for this loss depends on the heuristic.

Our results also indicate that there is a lot of potential for future improvement.

### 7.1 Other Cake-Cutting Algorithms

Our present experiment focuses on the Even-Paz algorithm since it provides both good fairness guarantees and a reasonable run-time complexity. However, there are other cake-cutting algorithms, which may fare better in some metrics.

### 7.2 Different Noise Patterns

We have seen that different noise pattern significantly effect the results of the tested algorithms. Thus, an important future work project is to test which noise pattern best reflects the differing valuations of real people for land.

### 7.3 Different Manipulation Strategies

Our results suggests a relationship between the risk of losing from acting strategic and the prospect of a high reward. Studying the risk/reward ratio of different agent manipulation strategies in future work provides more insights on the vulnerability of different cake-cutting algorithms to strategy and the profitability of acting strategic.

## References

Aumann Y, Dombb Y, Hassidim A (2013) Computing socially-efficient cake divisions. In: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, pp 343–350

Aziz H, Hassidim A, Segal-Halevi E (2020) Fair allocation with diminishing differences. Journal of Artificial Intelligence Research 67:471–507

Brams S, Denoon D (1997) Fair division: A new approach to the spratly islands controversy. International Negotiation 2(2):303–329

Brams S, Taylor A (1996) Fair division: From cake-cutting to dispute resolution. Cambridge University Press

Brams S, Togman J (1996) Camp David: Was the agreement fair? Conflict Management and Peace Science 15(1):99–112, DOI 10.1177/073889429601500105, URL http://dx.doi.org/10.1177/073889429601500105

Brams SJ (2007) Mathematics and Democracy: Designing Better Voting and Fair-Division Procedures, first edition edn. Princeton University Press, URL http://www.sciencedirect.com/science/article/pii/S089571770800174X

Brânzei S (2015) Computational Fair Division. PhD thesis, Faculty of Science and Technology in Aarhus university

Budish E, Cantillon E (2007) Strategic behavior in multi-unit assignment problems: Theory and evidence from course allocations. Computational Social Systems and the Internet 1(6.7):2007

Budish E, Cantillon E (2012) The multi-unit assignment problem: Theory and evidence from course allocation at harvard. American Economic Review 102(5):2237–71

Cavallo R (2012) Fairness and welfare through redistribution when utility is transferable. In: Proceedings of the Conference on Artificial Intelligence (AAAI)

Daniel T, Parco J (2005) Fair, Efficient and Envy-Free Bargaining: An Experimental Test of the Brams-Taylor Adjusted Winner Mechanism. Group Decision and Negotiation 14(3):241–264

Dickerson J, Goldman J, Karp J, Procaccia A, Sandholm T (2014) The computational rise and fall of fairness. In: Proceedings of the Conference on Artificial Intelligence (AAAI)

Dupuis-Roy N, Gosselin F (2011) The Simpler, the Better: A New Challenge for Fair-Division Theory. In: Proceedings of the Annual Meeting of the Cognitive Science Society, pp 3229–3234, URL http://csjarchive.cogsci.rpi.edu/Proceedings/2011/papers/0742/paper0742.pdf

Engelmann D, Strobel M (2004) Inequality aversion, efficiency, and maximin preferences in simple distribution experiments. American Economic Review 94(4):857–869

Even S, Paz A (1984) A Note on Cake Cutting. Discrete Applied Mathematics 7(3):285–296, DOI 10.1016/0166-218x(84)90005-2, URL http://dx.doi.org/10.1016/0166-218x(84)90005-2

Fehr E, Naef M, Schmidt K (2006) Inequality aversion, efficiency, and maximin preferences in simple distribution experiments: Comment. American Economic Review 96(5):1912–1917

Flood M (1958) Some experimental games. Management Science 5(1):5–26

Gal YK, Mash M, Procaccia AD, Zick Y (2017) Which is the fairest (rent division) of them all? Journal of the ACM (JACM) 64(6):39

Goldman J, Procaccia AD (2015) Spliddit: Unleashing fair division algorithms. ACM SIGecom Exchanges 13(2):41–46

Herreiner DK, Puppe C (2010) Inequality aversion and efficiency with ordinal and cardinal social preferencesan experimental study. Journal of Economic Behavior & Organization 76(2):238–253

Hortala-Vallve R, Llorente-Saguer A (2010) A simple mechanism for resolving conflict. Games and Economic Behavior 70(2):375–391, DOI 10.1016/j.geb.2010.02.005, URL http://dx.doi.org/10.1016/j.geb.2010.02.005

Kyropoulou M, Ortega J, Segal-Halevi E (2019) Fair cake-cutting in practice. In: Proceedings of the 2019 ACM Conference on Economics and Computation, ACM, pp 547–548

Lindner C, Rothe J (2016) Cake-cutting: Fair division of divisible goods. In: Rothe J (ed) Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 395–491

Massoud T (2000) Fair division, adjusted winner procedure (AW), and the Israeli-Palestinian conflict. Journal of Conflict Resolution 44(3):333–358, DOI 10.1177/0022002700044003003, URL http://dx.doi.org/10.1177/0022002700044003003

Mattei N, Walsh T (2013) Preflib: A library for preferences http://www. preflib. org. In: International Conference on Algorithmic DecisionTheory, Springer, pp 259–270

Menon V, Larson K (2017) Deterministic, strategyproof, and fair cake cutting. In: Proceedings of the Joint Conference on Artificial Intelligence (IJCAI)

Oluwasuji OI, Malik O, Zhang J, Ramchurn SD, et al. (2018) Algorithms for fair load shedding in developing countries. In: IJCAI, pp 1590–1596

Parco J, Rapoport A (2004) Enhancing honesty in bargaining under incomplete information: An experimental study of the bonus procedure. Group Decision and Negotiation 13(6):539–562

Procaccia A (2016) Cake cutting algorithms. In: Brandt F, Conitzer V, Endriss U, Lang J, Procaccia A (eds) Handbook of Computational Social Choice, Cambridge University Press, pp 311–330, DOI 10.1017/CBO9781107446984.014

Robertson J, Webb W (1998) Cake-cutting algorithms: Be fair if you can. AK Peters/CRC Press

Schneider G, Krämer U (2004) The limitations of fair division. Journal of Conflict Resolution 48(4):506–524, DOI 10.1177/0022002704266148, URL http://kops.ub.uni-konstanz.de/bitstream/handle/urn:nbn:de:bsz:352-opus-24028/The_limitation_of_fair_division_Kraemer_Schneider_JCR.pdf?sequence=1

Shapley L, Scarf H (1974) On cores and indivisibility. Journal of Mathematical Economics 1(1):23–37, DOI 10.1016/0304-4068(74)90033-0, URL https://doi.org/10.1016/0304-4068(74)90033-0

Steinhaus H (1948) The problem of fair division. Econometrica 16(1):101–104

Woeginger GJ, Sgall J (2007) On the complexity of cake cutting. Discrete Optimization 4(2):213–220

Itay Shtechman
The Open University of Israel
Email: itay.shtechman@gmail.com

Rica Gonen
The Open University of Israel
Email: ricagonen@gmail.com

Erel Segal-HaLevi
Ariel University, Ariel 40700, Israel
Email: erelsgl@gmail.com

# Appendix

## 1 Hot Spot Noise Pattern

For each variant, a map land-block coordinate $(x_c, y_c)$ was selected uniformly at random. The goal was to represent a subjective preference for an agent specific area of interest. The point $(x_c, y_c)$ represents a spot that the agent finds particularly valuable; the agent's valuation of other points is larger the closer they are to this "hot spot". The noise distance is limited using $r$. Formally, we set the value for each land-block using a function for a right circular cone were the x and y axes are the width and height of the map and the z-axis is the agent's interest (added noise):

$$z_{cone}(x, y) = h + \sqrt{cot(\theta)} * \sqrt{tan^2(\theta) * ((x_c - x)^2 + (y_c - y)^2)}$$

Cone height $h$ is chosen to be the shortest dimension of the map and $\theta := \arctan(1/4)$. We then set each land-block $(x_j, y_j)$ value to $(1 + HSN(x_j, y_j, r))V$, where $V$ is taken from the land-value map and $HSN$ is $z_{cone}$ normalized by the top of the cone and truncated by $r$:

$$HSN(x_j, y_j, r) = \begin{cases} z_{cone}(x_j, y_j)/z_{cone}(x_c, y_c), & \text{if } HSN > 1 - r \\ 0 \quad , & \text{Otherwise} \end{cases}$$

## 2 Find Optimal Cut Sequence Algorithm

As part of our proposed dynamic programming approach we present the *FindOptimalCut-Sequence (FOCS)* algorithm for an $n$ which is a power of 2 (Algorithm 1). The proof that FOCS is optimal uses the following lemma:

Denote by $X_C$ the division of cake $C$ to $n$ agents such that sub-cake $C_1$ is allocated to agents $a_1, ..., a_{n/2}$, and sub-cake $C_2$ to agents $a_{n/2+1}, ..., a_n$. Denote the rest of the division of $C_1$ and $C_2$ by $X_{C_1}$ and $X_{C_2}$ respectively. For every metric $m \in \{UV, EV\}$ there exists a monotonic function $f_m$ such that $m(X_C) = f_m(m(X_{C_1}), m(X_{C_2}))$.[1]

We show for every metric $m \in \{UV, EV\}$ that such function exists:

(i) $\mathrm{UV}(X_C) = \sum_{i=1}^{n} \frac{V_i(X_i)}{V_i(C)} = \sum_{i=1}^{n/2} \frac{V_i(X_i)}{V_i(C)} + \sum_{i=n/2+1}^{n} \frac{V_i(X_i)}{V_i(C)} = \mathrm{UV}(X_{C_1}) + \mathrm{UV}(X_{C_2})$

(ii) $\mathrm{EV}(X_C) = n \min_{1 \le i \le n} \frac{V_i(X_i)}{V_i(C)} = 2 \min(\mathrm{EV}(X_{C_1}), \mathrm{EV}(X_{C_2}))$

Given any cake $C$, a set of $n = 2^k$ agents (for some $k \in \mathbb{Z}_+$) and a metric function $m \in \{UV, EV\}$, the cake division made by the optimal cut (Horizontal or Vertical) sequence for metric $m$ amongst the $2^{n-1}$ options, is given by $FOCS(C, n, m)$. Let $X_{mnC}$ be the division to $n$ agents of cake $C$ made by the optimal cut sequence for metric $m \in \{UV, EV\}$. We will prove by induction that:

$$FOCS(C, n, m) = X_{mnC} \tag{1}$$

**Base case**: when $n = 2$, the algorithm tries all two possible cut sequences and chooses the best one, so (1) is true for $n = 2$.

**Induction step**: Let $k \in \mathbb{Z}_+$ be given and suppose (1) is true for $n = 2^k$. Then, for $n = 2^{k+1}$, consider two possible cut sequences:

[1] Since $LE$ computation depends on the relations between all pairs of agents, no such function exists for this metric.

**ALGORITHM 1:** Find Optimal Cut Sequence

**Input:** A cake $C$, a set of $n = 2^k$ agents, and a preferred metric function $m$.
**Output:** The allocation of cake $C$ to the $n$ agents.
**if** $n == 1$ **then**
  return $C$ as the agent's entire allocation;
**else**
  Each agent $i$ makes a vertical half-mark $x_i$. Order marks from left to right and renumber the agents so $x_1 < \ldots < x_n$. Let $x_M := (x_{n/2} + x_{n/2+1})/2$. Let $C_{left}, C_{right}$ be the halves of $C$ to the left and right of $x_M$;
  Set $VerticalAllocation =$
  $FOCS(C_{left},\ Agents\ 1 \ldots n/2,\ m) \cup FOCS(C_{right},\ Agents\ n/2+1 \ldots n,\ m)$ ;
  Each agent $i$ makes a horizontal half-mark $y_i$. Order marks from top to bottom and renumber the agents so $y_1 < \ldots < y_n$. Let $y_M := (y_{n/2} + y_{n/2+1})/2$. Let $C_{top}, C_{bottom}$ be the halves of $C$ to the top and bottom of $y_M$;
  Set $HorizontalAllocation =$
  $FOCS(C_{top},\ Agents\ 1 \ldots n/2,\ m) \cup FOCS(C_{bottom},\ Agents\ n/2+1 \ldots n,\ m)$ ;
  **if** $m(HorizontalAllocation) > m(VerticalAllocation)$ **then**
    return $HorizontalAllocation$;
  **else**
    return $VerticalAllocation$;
  **end**
**end**

(i) $X_{C_{hor}}$: A sequence starting with a horizontal cut dividing $C$ into $C_{top}$ and $C_{bottom}$.

(ii) $X_{C_{ver}}$: A sequence starting with a vertical cut dividing $C$ into $C_{left}$ and $C_{right}$.

By the induction hypothesis, the optimal sequence for dividing each sub-cake $C' \in \{ C_{top}, C_{bottom}, C_{left}, C_{right}\}$ among $n/2$ agents can be found using $FOCS(C', 2^k, m)$ — we denote those respective allocations by $X_{C_{top}}$, $X_{C_{bottom}}$, $X_{C_{left}}$, $X_{C_{right}}$.

Finally, from our lemma, we can use the function $f_m$ to measure $X_{C_{hor}}$ and $X_{C_{ver}}$ according to metric $m$ and choose the best one:

$$m(X_{C_{hor}}) = f_m(X_{C_{top}}, X_{C_{bottom}}) \quad ; \quad m(X_{C_{ver}}) = f_m(X_{C_{left}}, X_{C_{right}})$$

Thus, (1) holds for $n = 2^{k+1}$ too.

In each round, FOCS asks $2n$ queries and makes 4 recursive calls, so its run-time is given by the recurrence $T(n) = 2n + 4T(n/2); \quad T(1) = 0$, which gives $T(n) = \Theta(n^2)$. This is much better than exhaustive search, but may still be slow for large $n$. To allow for trade-off between run-time and performance in the defined metrics, we suggest various heuristics for selecting the cut-direction in each round. With these heuristics, it is not required to run 4 recursive calls, but only 2 recursive calls as in Even-Paz, so the run-time becomes $O(n \log n)$ as in Even-Paz.

# 3 Locally Optimal Strategy

The strategy of Alice for selecting her query-marks is calculated in two steps, at the first step Alice forces a better cut location in both possible cut directions using the following strategy:

1. Calculate the vertical and horizontal query-marks of all other $n$ agents (this does not require to assume that Alice knows the entire valuation functions of the other agents — only that she knows their half-points).

2. If Alice's mark is among the leftmost (or bottom) $n/2$ agents, then she selects a mark slightly to the left (or below) of the mark number $n/2 + 1$ from the left (or bottom).

3. Otherwise, Alice's mark is among the rightmost (or top) $n/2$ agents; then she selects a mark slightly to the right (or above) of the mark number $n/2$ from the left (or bottom).

At the second step of her strategic selection, if the cutting heuristic is not a predefined one, Alice tries forcing the cut direction to further expand her local gain using the following strategy:

1. Calculate the selected cut direction $D_{sel}$ according to the current cut heuristic and the newly selected query-marks. Additionally, calculate the potential cut locations for both horizontal and vertical cuts.

2. If Alice's gain from cutting in $D_{sel}$ direction is less then her possible gain from cutting in the opposite direction $D_{opp}$, Alice searches for a different $D_{sel}$ direction query-mark $q$ that will force the current heuristic to select $D_{opp}$ as the cut direction.

3. If such $q$ was found, Alice selects her new query-mark for $D_{sel}$ direction to be $q$, otherwise, Alice does not change her already selected strategic query-marks.

Using this strategy, Alice remains in the same group, while maximizing the piece given to her own group of $n/2$ agents. In some cases, depending on the cut heuristic and the other agents, Alice also influences the cut direction by intentionally selecting a bad query-mark in the direction she is less interested in.

# 4 Figures

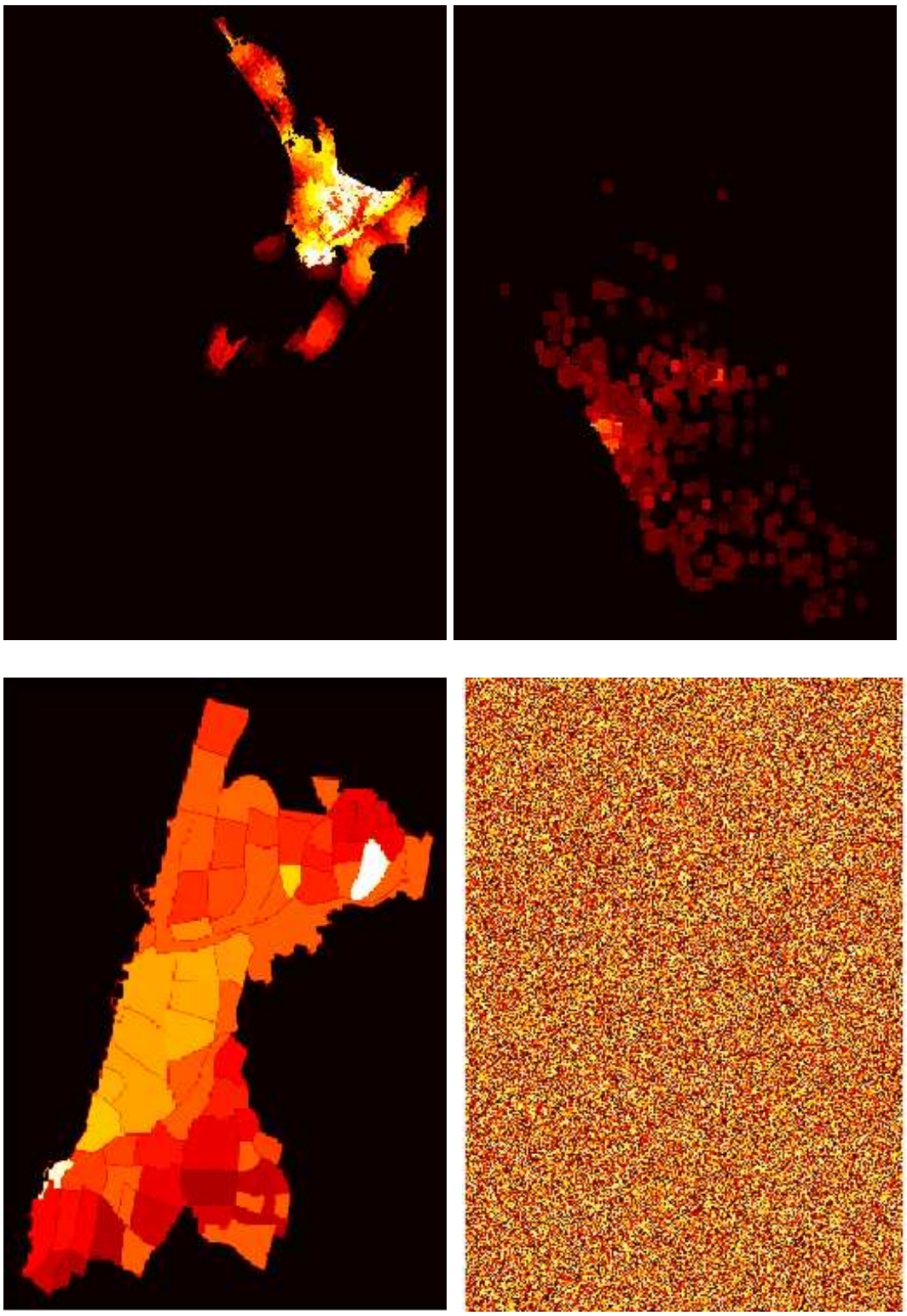

Figure 1: Heat maps based on the four land-value maps (top left- New Zealand; top right- Israel; bottom left- Tel-Aviv; bottom right- Random Noise)

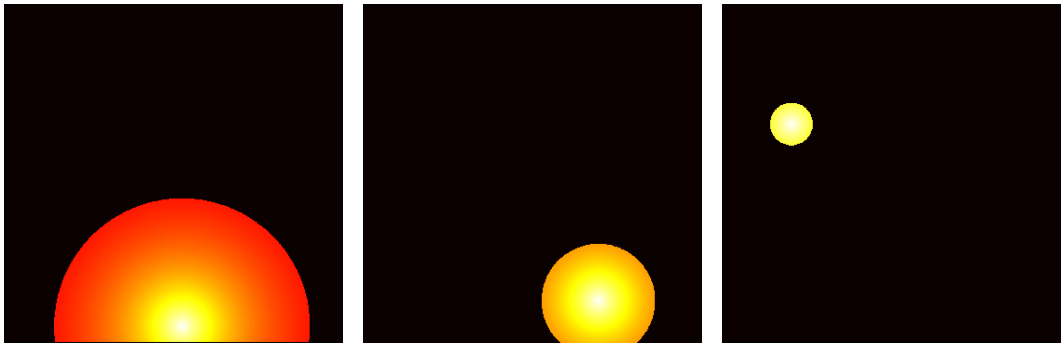

Figure 2: Heat maps illustrating the hot spot noise pattern (left to right - r=60%,r=40%,r=20%)

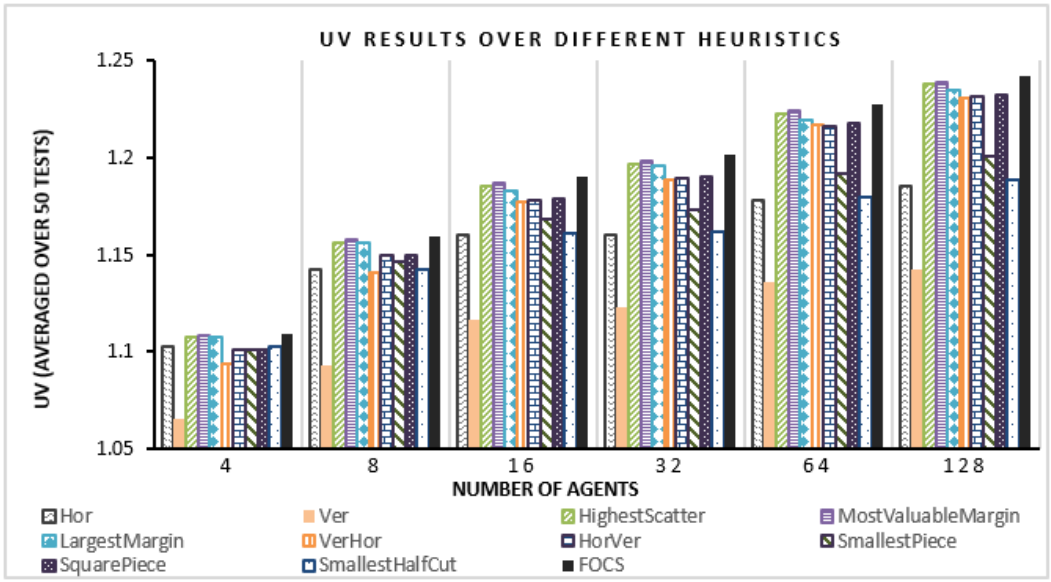


Figure 3: Utilitarian-value of Even-Paz with various heuristics for New Zealand dataset created by hotspot noise pattern (the graph for uniform noise pattern is similar).

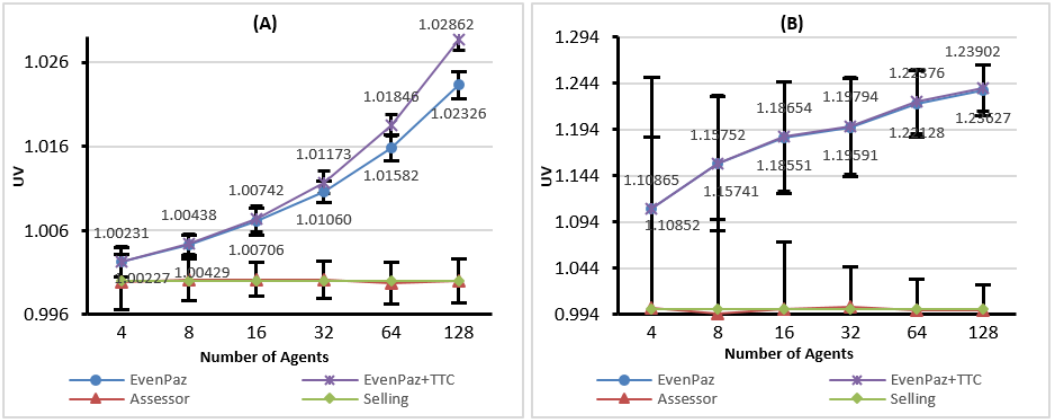


Figure 4: Utilitarian-value of Even-Paz with MostValuableMargin before and after TTC compared to UV of an assessor division and market sale. The vertical lines denote 95% confidence intervals.
(A) Uniform noise pattern (B) Hotspot noise pattern

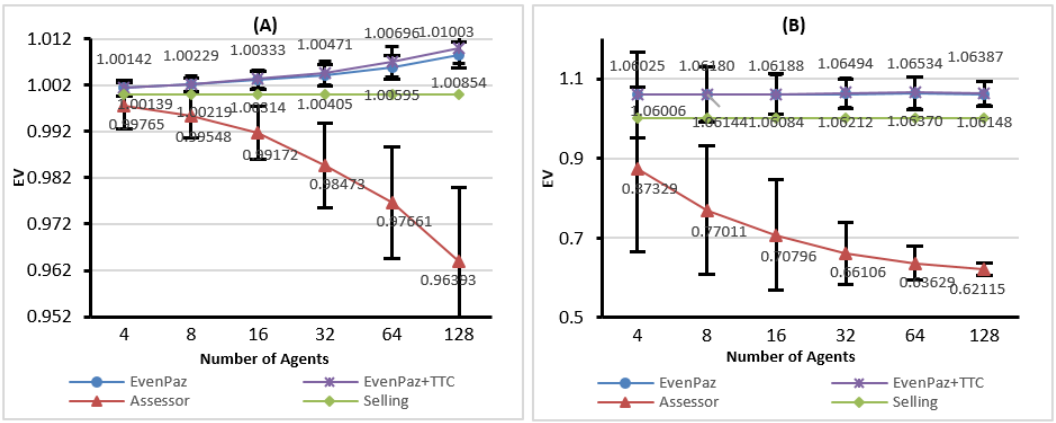


Figure 5: Egalitarian-value of Even-Paz with MostValuableMargin before and after TTC compared to EV of an assessor division and market sale. The vertical lines denote 95% confidence intervals.
(A) Uniform noise pattern (B) Hotspot noise pattern

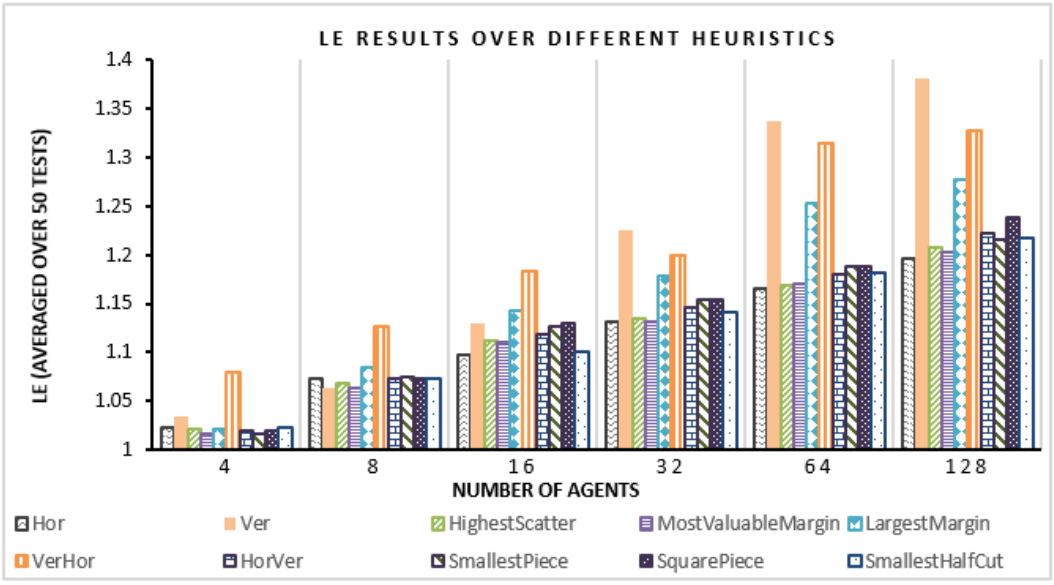


Figure 6: Largest-envy of Even-Paz with various heuristics. New Zealand map, hotspot noise.

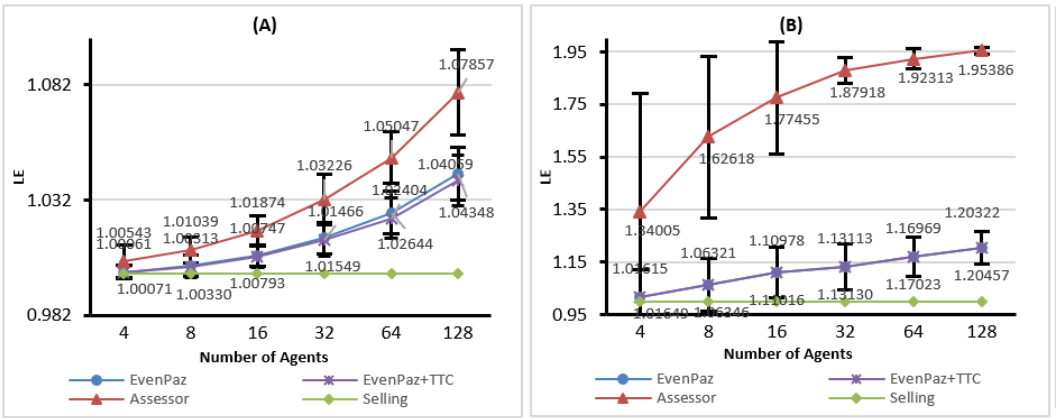


Figure 7: Largest-envy of Even-Paz with MostValuableMargin compared to LE of an assessor division and market sale. The vertical lines denote 95% confidence intervals. (A) Uniform noise pattern (B) Hotspot noise pattern