Fair division algorithms with a small number of queries

Andrei Malyutin

Abstract. We study the algorithmic complexity of fair division problems with a focus on minimizing the number of queries needed to find an approximate solution with desired precision. In a recent joint work with Alexandr Grebennikov, Xenia Isaeva, Mikhail Mikhailov, and Oleg Musin, we showed for several classes of fair division problems that under certain natural conditions on sets of preferences, a polylogarithmic number of queries with respect to the reciprocal of accuracy is sufficient. The present note extends these results (on the sufficiency of polylogarithmic number of queries) to the case of four or more tenants in the rental harmony problem with convex preference sets.

We study algorithmic aspects of the so-called fair division problems. A nice introduction to the subject is given in the book [RW98]. In this note, we discuss the following specific algorithmic geometry problem, the relation of which to the rental harmony problem (this is a type of fair division problems) is explained, e. g., in a recent paper [GIMMM].

1. Stating an algorithmic problem

Let $k \geq 2$ be a positive integer, let Δ_k be a $(k-1)$-dimensional regular simplex with edges of length 1 in $\mathbb{R}^{k-1}$, and let $v_1, \ldots, v_k$ be the vertices of Δ_k. For $j \in \{1, \ldots, k\}$, we denote the facet $\mathrm{Conv}\,(\{v_i\}_{i \neq j})$ of Δ_k, where $\mathrm{Conv}\,(X)$ stands for the convex hull of X, by F_j. (For the rental harmony problem, Δ_k corresponds to all representations of total price as a sum of k nonnegative numbers; and F_j is precisely the set of price distributions with zero price for the jth room.)

Assume that a collection of k subsets $P_1, \ldots, P_k$ of Δ_k is fixed such that

(P1) $\{P_1, \ldots, P_k\}$ is a covering of Δ_k, that is, $\bigcup_{i \in \{1,\ldots,k\}} P_i = \Delta_k$;
(P2) P_i contains F_i for each $i \in \{1, \ldots, k\}$;

This work was supported by RFBR, research project no. 20-01-00070.

(P3) P_i is convex for each $i \in \{1, \dots, k\}$;
(P4) P_i is closed for each $i \in \{1, \dots, k\}$.[1]

Assume that we have no description of the sets $P_1, \dots, P_k$, but we know that the sets have the listed properties and we can perform queries about these sets: if we choose an index $i \in \{1, \dots, k\}$ and a point $x \in \Delta_k$, we receive 'yes' if $x \in P_i$ and 'no' otherwise.

The Knaster–Kuratowski–Mazurkiewicz (KKM) lemma guarantees (due to the properties (P1), (P2), and (P4)) that the intersection $\bigcap_{i \in \{1,\dots,k\}} P_i$ is nonempty, so that there exists a point x in Δ_k such that $\operatorname{dist}(x, P_i) = 0$ for all $i \in \{1, \dots, k\}$. We say that such x in Δ_k is a *solution* for $\mathscr{P} = \{P_1, \dots, P_k\}$. We say that x in Δ_k is an *ε-solution*, $\varepsilon \geq 0$, for $\mathscr{P}$ if $\operatorname{dist}(x, P_i) \leq \varepsilon$ for all $i \in \{1, \dots, k\}$.[2]

Our goal is to construct an algorithmic procedure that, given a collection $\{P_1, \dots, P_k\}$ (having properties (P1)–(P4)) and an 'accuracy constant' $\varepsilon > 0$ finds an ε-solution using the smallest possible number of queries.

2. The main result

It is known that ε-nets allow us to find ε-solutions with $O(1/\varepsilon)^{k-1}$ queries. In the case $k = 2$, we can use binary search to find an ε-solution in the interval Δ_2 with $O(\log(1/\varepsilon))$ queries. Theorem 5.2 in [GIMMM] implies that in the case $k = 3$, an ε-solution can be found with $O(\log(1/\varepsilon))^2$ queries. A natural conjecture arises that an ε-solution can be found with $O(\log(1/\varepsilon))^{k-1}$ queries. The main result of the present note confirms this conjecture for $k = 4$.

Theorem 1. *In the case $k = 4$, an ε-solution for convex P_i can be found with $O(\log(1/\varepsilon))^3$ queries.*

Theorem 1 implies (modulo results of [GIMMM] and using terminology introduced there) that, for the rental harmony problem with 4 tenants having convex preference sets, we can find an ε-fair division point in binary mode with $O(\log(1/\varepsilon))^3$ queries.

3. Basic idea of the algorithm

The description of our algorithm is rather cumbersome, and before proceeding to it, we will outline its core idea. This idea seems to work for an arbitrary dimension, but here we restrict ourselves to the case of $k = 4$. Let us fix one of the facets F_1, F_2, F_3, F_4 (say, F_4) and study the sections of our Δ_4 with hyperplanes parallel to F_4. These sections form a bundle $(T_\theta)_{\theta \in [0,1]}$ of regular triangles, where θ is the diameter of T_θ; we have $T_0 = v_4$ and $T_1 = F_4$. The first key observation we need

[1] In fact, we can carry out our constructions without condition (P4), but we introduce it for convenience, in order not to repeatedly mention the transition to closures in what follows.
[2] We remark that an ε-solution is not necessarily located close to a solution. It is easy to construct an example when the distance between an ε-solution and the solution closest to it exceeds 100ε.

is that if P_1, P_2, P_3 do not cover T_θ for some θ (so that $T_\theta \setminus (P_1 \cup P_2 \cup P_3)$ is nonempty) then T_θ contains a unique '*inscribed circle*' S_θ that touches each of P_1, P_2, and P_3 and has no points of $P_1 \cup P_2 \cup P_3$ inside. The center of S_θ is an r_θ-solution, where r_θ is the radius of S_θ. Thus, if given an $\varepsilon > 0$ we find some $\theta \in [0,1]$ such that $T_\theta \setminus (P_1 \cup P_2 \cup P_3)$ is nonempty and $r_\theta \leq \varepsilon$, then the center of S_θ will be the desired ε-solution. Observe that, due to convexity arguments, if the set $\mathscr{I}$ of those θ for which P_1, P_2, P_3 do not cover T_θ is nonempty, then this $\mathscr{I}$ is a half-open subinterval in $[0,1]$ of the form $(\theta',1]$, and the radii r_θ form a continuous monotone function on this subinterval $\mathscr{I}$. Using this, we can try to find θ with a small radius $r_\theta \leq \varepsilon$ via binary search. Take $\theta_1 = 1/2$. If $1/2$ is not in $\mathscr{I}$, then put $\theta_2 = 3/4$, and if $1/2$ is in $\mathscr{I}$ and $r_{1/2} > \varepsilon$, then put $\theta_2 = 1/4$, and so on. For example, if the set $\mathscr{I}$ is empty or of length less than ε, then at the step $m = \log_2\lceil 1/\varepsilon\rceil$ we get $\theta_m = 1 - 1/2^m \geq 1 - \varepsilon$ and any point in the nonempty (by the KKM lemma) set $T_{\theta_m} \cap P_1 \cap P_2 \cap P_3$ is an ε-solution (because $F_4 \subset P_4$ is close enough).

A difficulty arising when implementing the described idea as an algorithmic procedure is that a monotone function can have 'jumps'. This issue is resolvable due to the fact that the function r_θ is convex in addition. Another difficulty is the calculation of inscribed circles.

3.1. Inscribed antitriangles

In order to simplify computations, instead of finding (approximately) inscribed circles S_θ for triangles T_θ, we introduce and calculate (approximately) *inscribed antitriangles*. By an *antitriangle* in a triangle T_θ (in the above notation) we mean any regular triangle contained in T_θ that is related to T_θ by a negative homothetic transformation. An antitriangle A in T_θ is *inscribed* if the intersection of A with the union $P_1 \cup P_2 \cup P_3$ is the set of vertices of A. It can be shown that if $T_\theta \setminus (P_1 \cup P_2 \cup P_3)$ is nonempty then there exists a unique inscribed antitriangle A_θ for T_θ. Observe that if $T_\theta \setminus (P_1 \cup P_2 \cup P_3)$ is nonempty, then the center of the inscribed antitriangle A_θ is a $(\mathrm{Diam}\,(A_\theta)/\sqrt{3})$-solution, where $\mathrm{Diam}\,(A_\theta)$ is the diameter of A_θ, while any point in A_θ is a $\mathrm{Diam}\,(A_\theta)$-solution. Similar to the approach with inscribed circles, the diameters $\mathrm{Diam}\,(A_\theta)$ form a continuous monotone convex function on $\mathscr{I}$. Using this, we can try to find θ with a small $\mathrm{Diam}\,(A_\theta)$ via binary search.

4. Procedure for finding inscribed antitriangles (PFIA)

Now we describe a computational procedure that, given an arbitrary $\theta \in (0,1]$, operates in the triangle T_θ and calculates, with a prescribed precision, the size of the inscribed antitriangle A_θ and its position (if it is large enough). The input of the procedure is the 'coordinate' θ and a 'precision constant' $\delta > 0$. The outputs of the procedure are:

- A (nonnegative real) number d_θ such that $|d_\theta - \mathrm{Diam}\,(A_\theta)| \leq \delta$ (if A_θ is undefined, we formally set $\mathrm{Diam}\,(A_\theta) = 0$ so that $d_\theta \leq \delta$ in this case).

- A point x_θ in T_θ such that the metric ball $B_{d_\theta+\delta}(x_\theta)$ of radius $d_\theta+\delta$ centered at x_θ intersects P_1, P_2, and P_3. Besides, $B_{d_\theta+\delta}(x_\theta)$ intersects P_4 whenever $\mathrm{Diam}\,(A_\theta) > \delta$.

In this procedure, we fix one of the facets F_1, F_2, F_3 (say, F_3) and regard T_θ as a bundle of closed segments parallel to the edge $T_\theta \cap F_3$. For this bundle, we use the notation $(I_\alpha)_{\alpha\in[0,\theta]}$, where I_α is the segment of length α (so that $I_\theta = T_\theta \cap F_3$ and I_0 is the opposite vertex of T_θ). For each $\alpha \in [0,\theta]$, the segment I_α is isometric to the segment $[0,\alpha]$. In the following description, when α is fixed we identify I_α with $[0,\alpha]$ via the isometry sending the endpoint $I_\alpha \cap F_1$ to $\{0\}$ in $[0,\alpha]$.

If $T_\theta \setminus (P_1 \cup P_2 \cup P_3)$ is nonempty and the inscribed antitriangle A_θ exists, then there is a unique $\eta \in (0,\theta)$ such that I_η contains an edge of A_θ. In this case we use the notation $E(\theta) = \eta$. If $T_\theta \setminus (P_1 \cup P_2 \cup P_3)$ is empty, we set $E(\theta) = \theta$. The procedure uses several levels of binary searches, the upper level goes through the interval $[0,\theta]$ and study segments I_α of $(I_\alpha)_{\alpha\in[0,\theta]}$, which can be regarded as aiming to 'find' (approximately) the segment $I_{E(\theta)}$.

We set $a_1 = 0$ and $b_1 = \theta$ and start an iterative process with intervals $[a_i, b_i]$ in $[0,\theta]$ such that $[a_i, b_i]$ contains $E(\theta)$ if $\mathrm{Diam}\,(A_\theta)$ is large enough. Given a_i and b_i such that $0 \le a_i < b_i \le \theta$, the ith iteration looks as follows. We set $c_i := (a_i + b_i)/2$ and operate in the segment I_{c_i} (which is parametrized as $[0, c_i]$ by the above convention). In $m := \lceil \log_2(9c_i/\delta)\rceil \le \lceil \log_2(1/\delta)\rceil + 4$ queries we can find in $I_{c_i} = [0, c_i]$ a half-open subinterval of the form $[p\delta', p\delta' + \delta')$, where p is an integer and $\delta' = c_i/2^m \le \delta/9$, such that

$$[0, p\delta'] \subset (I_{c_i} \cap P_1) \subset [0, p\delta' + \delta').$$

Another m queries allow us to find $q \in \mathbb{Z}$ such that $(q\delta', q\delta' + \delta']$ contains the endpoint g of $I_{c_i} \cap P_2$ such that $[g, c_i] = I_{c_i} \cap P_2$. We have three cases:

- $q < p$, which means that $P_1 \cup P_2$ contains I_{c_i} and $c_i < E(\theta)$. In this case, we set $[a_{i+1}, b_{i+1}] := [c_i, b_i]$ and pass to the next iteration.
- $q = p$, which means that either $P_1 \cup P_2$ contains I_{c_i} or the interval $I_{c_i} \setminus (P_1 \cup P_2)$ has length at most δ. In the case $q = p$, we also set $[a_{i+1}, b_{i+1}] := [c_i, b_i]$ for the next iteration, even though it is possible that $E(\theta) = c_i$. In fact, if $q = p$, then $E(\theta)$ can take any position in $(0, c_i)$, but we see that if $q = p$ and $E(\theta) \le c_i$, then the diameter of A_θ is at most δ, which is less than our 'level of visibility' limit.
- $q > p$, in this case we see that the length of $I_{c_i} \setminus (P_1 \cup P_2)$ is greater than $(q-p-1)\delta'$ and lesser than $(q-p+1)\delta'$. In the case $q > p$, we do additional computations.

Additional computations for the case $q > p$ are as follows. If the subsegment $[p\delta', q\delta' + \delta']$ of I_{c_i} is an edge of an antitriangle contained in T_θ, we denote the opposite vertex of this antitriangle by w. If the subsegment $[p\delta' + \delta', q\delta']$ of I_{c_i} is an edge of an antitriangle contained in T_θ, we denote the opposite vertex of this antitriangle by v. If any of v and w is defined, we perform a query whether P_3 contains it. Then, for the case $q > p$, we introduce three subcases:

(L) either v is not defined or $v \in P_3$. In this case, we have $E(\theta) < c_i$, and we set $[a_{i+1}, b_{i+1}] := [a_i, c_i]$ (and pass to the next iteration).
(R) w is defined and $w \notin P_3$. In this case, $E(\theta) > c_i$. We set $[a_{i+1}, b_{i+1}] := [c_i, b_i]$.
(+) v is defined and $v \notin P_3$ while w is either not defined or $w \in P_3$. In this case, we have $|\mathrm{Diam}\,(A_\theta) - (q-p)\delta'| \le \delta'$, and we stop our procedure with setting $d_\theta = (q-p)\delta'$ and x_θ to be the center of the antitriangle with edge $[(p+1)\delta', q\delta']$ (or just set x_θ to be any point of this edge).

This completes the description of the iterative step.

We continue the iterative process either until subcase (+) happens or stop at step $2\lceil \log_2(1/\delta)\rceil + 10$. If subcase (+) happens, the output of the procedure is described above. If we stop at the step $t = 2\lceil \log_2(1/\delta)\rceil + 10$ with no (+) subcase, the situation splits in two following subcases.

- The subcase with $a_t > \theta - \delta/2$. It can be shown that the only way to get $a_t > \theta - \delta/2$ is to have 'short' interval $I_{\theta-\delta} \setminus (P_1 \cup P_2)$ of length less than $\delta + 2\delta/9$. In this case we have $\mathrm{Diam}\,(A_\theta) < 2\delta$ and we can set $d_\theta = \delta$. The point x_θ can be chosen in $I_{\theta-\delta}$ in an obvious way.
- If $a_t \le \theta - \delta/2$, we study the segment I_{a_t}. Clearly, $2\lceil \log_2(1/\delta)\rceil + 10$ queries is enough to find h such that
$$|h - \mathrm{Diam}\,(I_{a_t} \setminus (P_1 \cup P_2))\,| < \delta/9$$
and $x \in I_{a_t}$ such that x $(\delta/9)$-approximates the center of that of the two intervals $I_{a_t} \setminus (P_1 \cup P_2)$ and $I_{a_t} \cap P_1 \cap P_2$ which is nonempty. Then we set $d_\theta := h + \delta/2$ and $x_\theta := x$ and quit the procedure.[3]

5. Description of the algorithm (of finding an ε-solution)

Now we turn to the description of our algorithm that, given an 'accuracy constant' ε and a collection $\mathscr{P} = \{P_1, P_2, P_3, P_4\}$ of subsets with properties (P1)–(P4) in Δ_4, following the basic idea described above, and using the procedure described above (PFIA), finds an ε-solution for this $\mathscr{P}$.

Algorithm starts with applying PFIA to the triangle $T_{1-\varepsilon}$ (see notation in Sec. 3). Let the input accuracy constant δ for PFIA be $\varepsilon/9$.

If PFIA says that $d_{1-\varepsilon} \le \varepsilon - \delta$, which means that the set $T_{1-\varepsilon} \setminus (P_1 \cup P_2 \cup P_3)$ is either empty or 'thin enough', then the point $x_{1-\varepsilon}$ (this point is in the output of PFIA; see the description of PFIA) is an ε-solution because the metric ball $B_\varepsilon(x_{1-\varepsilon})$ of radius ε centered at $x_{1-\varepsilon}$ intersects P_4 (which contains F_4) and P_1, P_2, and P_3 as well (because $B_{d_{1-\varepsilon}+\delta}(x_{1-\varepsilon})$ by construction of PFIA intersects

[3] In order to prove that the assigned values of d_θ and x_θ indeed have the properties declared for the output, we check several various cases and use properties of convex sets. One of the key points of our proof is the fact that corresponding endpoints of the intervals $I_{a_t} \setminus (P_1 \cup P_2)$ and $I_{b_t} \setminus (P_1 \cup P_2)$ are located at a small distance from each other. This fact follows from the conditions $a_t \le \theta - \delta/2$ and $b_t - a_t \le \delta^2/100$ and can be proved by analogy with the simple observation given in Sec. 5.2.

P_1, P_2, and P_3 while $d_{1-\varepsilon}+\delta \le \varepsilon-\delta+\delta=\varepsilon$ so that $B_{d_{1-\varepsilon}+\delta}(x_{1-\varepsilon}) \subset B_\varepsilon(x_{1-\varepsilon})$). Then the algorithm stops.

Otherwise, if $d_{1-\varepsilon} > \varepsilon-\delta$, the algorithm goes into an iterative process with intervals $[a_i, b_i]$ in $[0,1]$ such that $d_{a_i} \le \delta$ and $d_{b_i} > \varepsilon-\delta$ (in the present settings we have $\varepsilon-\delta = 8\varepsilon/9$). Having $d_{1-\varepsilon} > \varepsilon-\delta$, we set $[a_1, b_1] = [2\delta, 1-\varepsilon]$. Each subsequent iteration, being given $[a_i, b_i]$ with $d_{a_i} \le \delta$ and $d_{b_i} > \varepsilon-\delta$, we set $c_i = (a_i+b_i)/2$ and apply PFIA to the triangle T_{c_i} obtaining d_{c_i} and x_{c_i} as its output.

- If $d_{c_i} \le \delta$, we move on to the next iteration with $[a_{i+1}, b_{i+1}] := [c_i, b_i]$.
- If $d_{c_i} > \varepsilon-\delta$, we move on to the next iteration with $[a_{i+1}, b_{i+1}] := [a_i, c_i]$.
- If $d_{c_i} \in (\delta, \varepsilon-\delta]$, then $\mathrm{Diam}\,(A_{c_i}) \in (0,\varepsilon]$ (because $|d_\theta - \mathrm{Diam}\,(A_\theta)\,| \le \delta$) and x_{c_i} is an ε-solution for $\mathscr{P}$ (by construction of PFIA).

5.1. Estimating the number of queries

Observe that $\mathrm{Diam}\,(A_t)$, $t \in [0,1]$, is a convex nonnegative function. In particular, for any a and b in $[0,1]$ such that $0 \le a < b \le 1$, we have (cf. Sec. 5.2)

$$\frac{\mathrm{Diam}\,(A_b) - \mathrm{Diam}\,(A_a)}{b-a} \le \frac{\mathrm{Diam}\,(A_1) - \mathrm{Diam}\,(A_a)}{1-a}.$$

Suppose that the upper level iterative process in our algorithm arrives at step i. Since $\mathrm{Diam}\,(A_1) \le 1/2$, $a_i < b_i \le 1-\varepsilon$, $|d_\theta - \mathrm{Diam}\,(A_\theta)\,| \le \delta$, and $\mathrm{Diam}\,(A_{a_i}) \le d_{a_i} + \delta \le 2\delta$ in our case, it follows that

$$d_{b_i} \le \frac{b_i - a_i}{2\varepsilon} + 3\delta.$$

Since $\varepsilon - \delta < d_{b_i}$ and $\delta = \varepsilon/9$, this implies that

$$\varepsilon^2 < b_i - a_i.$$

Since $b_i - a_i \le 2^{1-i}$, it follows that a necessary conditions for the transition to the ith iteration is the validity of the inequality

$$i < \log_2(2/\varepsilon^2) = 1 + 2\log_2(1/\varepsilon).$$

Therefore, since we refer to PFIA before iterations only once, our algorithm arrives at an ε-solution by calling procedure PFIA at most $1 + 2\log_2(1/\varepsilon)$ times. Each iteration of PFIA requires at most $(2\lceil\log_2(1/\varepsilon)\rceil + 10)^2$ queries, so the total search takes at most $(2\lceil\log_2(1/\varepsilon)\rceil + 10)^3$ ones.

5.2. An observation concerning convex/concave functions

Let $f\colon [0,1] \to \mathbb{R}$ be a nonnegative concave function with domain $[0,1]$, and let a and b be numbers in $[0,1]$ such that $0 \le a < b \le 1$. Then

$$\frac{f(b)-f(a)}{b-a} \le \frac{f(b)}{b} \quad \text{and} \quad \frac{f(a)-f(b)}{b-a} \le \frac{f(a)}{1-a}.$$

In particular, if $f(a)$ and $f(b)$ are in $[0,1]$ and for some $\delta > 0$ we have $\delta \le a < b \le 1-\delta$ and $b - a \le \delta^2$, then

$$|f(b) - f(a)| \le \delta.$$

References

[A18] Asada, M., Frick, F., Pisharody, V., Polevy, M., Stoner, D., Tsang, L., Wellner, Z.: Fair division and generalizations of Sperner- and KKM-type results. SIAM J. Discrete Math. **32**, 591–610 (2018)

[AZ14] Azrieli, Y., Shmaya, E.: Rental harmony with roommates. Journal of Mathematical Economics **46**, 303–310 (2014)

[Ba89] Bapat, R. B.: A constructive proof of a permutation-based generalization of Sperner's lemma. Mathematical Programming **44**, 113–120 (1989)

[DQS12] Deng, X., Qi, Q., Saberi, A.: Algorithmic Solutions for Envy-Free Cake Cutting. Operations Research **60**, 1461–1476 (2012)

[DS61] Dubins, L. E., Spanier, E. H.: How to cut a cake fairly. Amer. Math. Monthly **68**, 1–17 (1961)

[FHEM19] Frick, F., Houston-Edwards, K., Meunier, F.: Achieving rental harmony with a secretive roommate. Amer. Math. Monthly **126**(1), 18–32 (2019)

[Ga84] Gale, D.: Equilibrium in a discrete exchange economy with money. International Journal of Game Theory **13**, 61–64 (1984)

[GIMMM] Grebennikov, A., Isaeva, X., Malyutin, A. V., Mikhailov, M., Musin, O. R.: Logarithmic algorithms for fair division problems (2021). `arXiv:2112.13622`

[KKM29] Knaster, B., Kuratowski C., Mazurkiewicz S.: Ein Beweis des Fixpunktsatzes für n-dimensionale Simplexe. Fundamenta Mathematicae **14**, 132–137 (1929)

[Mu14] Musin, O. R.: Around Sperner's lemma (2014). `arXiv:1405.7513`

[Mu16] Musin, O. R.: Homotopy invariants of covers and KKM-type lemmas. Algebr. Geom. Topol. **16**(3), 1799–1812 (2016)

[Mu17] Musin, O. R.: KKM type theorems with boundary conditions. J. Fixed Point Theory Appl. **19**(3), 2037–2049 (2017)

[RW98] Robertson, J., Webb, W.: Cake-Cutting Algorithms: Be Fair if You Can (1st ed.). A K Peters/CRC Press, Nattick, MA (1998)

[Ste49] Steinhaus, H.: Sur las division pragmatique. Econometrica **17**, 315–319 (1949)

[Str80] Stromquist, W.: How to cut a cake fairly. Amer. Math. Monthly **87**, 640–644 (1980)

[Su99] Su, F. E.: Rental harmony: Sperner's lemma in fair division. *Amer. Math. Monthly* **106**(10), 930–942 (1999)

Andrei Malyutin
St. Petersburg Department of
Steklov Institute of Mathematics
St. Petersburg State University
St. Petersburg, Russia
e-mail: `malyutin@pdmi.ras.ru`