MSCI 700: Game Theory - Course Project
Tractable Variants of Envy-Free Cake Cutting

Sushant Agarwal

November 6, 2019

1 Introduction

1.1 History and Motivation of fair division

Informally, fair division consists of dividing a set of objects between several parties such that each person receives their due share.

This problem arises in various real-world settings such as divorce settlements, traffic management etc. Fairly dividing limited natural resources (such as fossil fuels, water etc.) is also very important.

The 20th century experienced a shift toward mathematically rigorous approaches to fairness.

Fair division is an active research area in Mathematics, Economics, Game theory and Computer Science. Much of the research has aimed to provide algorithms to achieve a fair division, or prove its impossibility.

As [7] mentions, the central tenet of fair division is that such a division should be performed by the players themselves, maybe using a mediator but certainly not an arbiter as only the players really know how they value the goods.

There are various kinds of fair division problems which depend on the nature of objects (homogeneous, heterogeneous) to divide, the criteria for fairness (proportionality, envy-freeness) and various other criteria. The literature also distinguishes between allocation of divisible goods such as land, time, and memory on a computer, and indivisible goods (such as a house or a computer).

Each variation has attracted ample attention. Computer scientists have focused mainly on the allocation of indivisible goods. I will be focusing on the fair allocation of heterogeneous divisible goods, particularly a useful framework commonly known as cake cutting.

1.2 History and Motivation of cake cutting

As [5] mentions, the setup is basically the following: We need to fairly divide a birthday cake among several children. The cake is heterogeneous: two pieces of cake may differ in terms of their toppings. The children have different tastes, so the children have different preferences over the pieces(one may prefer a larger proportion of chocolate curls, while another may desire the piece with the cherry).

Cake cutting is a useful metaphor for the task of allocating a heterogeneous divisible good among multiple players with different preferences. As [6] says, procedures for fair cake-cutting can be used to divide various kinds of resources, such as land estates, advertisement space or broadcast time.

Research on cake cutting traditionally focuses on either establishing the existence of cake divisions with desirable properties (such as envy-freeness, proportionality), or designing cake-cutting algorithms to help players achieve fair allocations. I will focus on the second.

Research in cake cutting can be successfully applied to important real-world problems. As [4] mentions, in their 1996 book, Brams and Taylor discussed at length how cake-cutting algorithms can be applied to high-stakes negotiations and to divorce settlements. Cake-cutting algorithms have also been commercialized by companies like Fair Outcomes, Inc. (http://www.fairoutcomes.com/) and Spliddit (http://www.spliddit.org).

2 Setup

The setting I consider is the same as the setting in [5]. The set of n players is denoted $N = \{1, ..., n\}$, and the cake is represented by the interval $[0, 1]$. Each player $i \in N$ has a unique valuation function V_i, which maps an interval $I \subseteq [0, 1]$ to the value assigned to it by player i, denoted by $V_i(I)$. We use $V_i(x, y)$ as a shorthand for $V_i([x, y])$. These valuation functions are assumed to the following conditions for every $i \in N$:

- *Normalization:* $V_i(0, 1) = 1$
- *Divisibility:* For every subinterval $[x, y]$ and $0 \leq \lambda \leq 1$ there exists a point $z \in [x, y]$ such that $V_i(x, z) = \lambda V_i(x, y)$. The divisibility property implies that the valuation functions are non-atomic, that is, $V_i(x, x) = 0$ for every $x \in [0, 1]$.
- *Non-negativity:* For every subinterval I, $V_i(I) \geq 0$.
- *Additivity:* For two disjoint subintervals I, I', $V_i(I) + V_i(I') = V_i(I \cup I')$.

A piece of cake is a finite union of disjoint intervals.

We are interested in allocations $A = (A_1, \ldots, A_n)$, where A_i is the piece of cake allocated to player i. These pieces are assumed to form a partition of the cake, which means that the pieces are disjoint and their union is the entire cake.

We consider the following fairness properties:

- *Proportionality:* for all $i \in N$, $V_i(A_i) \geq 1/n$.
- *Envy-freeness:* For all $i, j \in N$, $V_i(A_i) \geq V_i(A_j)$

Intuitively, proportionality implies that every player has value at least $1/n$ for its piece of cake. Envy-freeness implies that each player weakly prefers his own piece to any other piece.

Proportionality is implied by envy-freeness. In the case of n players, any partition of the cake into n pieces must include a piece worth at least $1/n$ according to given player's valuation function (by the pigeon-hole principle). Envy-freeness then implies the value of the player's own piece is at least $1/n$, which in turn implies proportionality.

However, the converse is not true, as shown in Section 3.2.

3 Some basic Algorithms

I discuss some simple cake cutting algorithms to give a brief introduction to some of the ideas used in these settings.

3.1 Cut and Choose

The first cake-cutting algorithm which I discuss is the cut-and-choose algorithm for fairly dividing a cake between two players. As mentioned in [4], the first player starts by dividing the cake into two pieces he values equally. The second player then picks the piece he values more, and the first player receives the other piece.

Cut and choose is guaranteed to produce a proportional allocation: each player receives a piece he values at $1/2$ the value of the whole cake. The algorithm produces an envy-free allocation: each player likes his own piece at least as much as the other piece.

It is easy to prove the above statements. It is enough to show that each player receives a piece he values at at least $1/2$. The first player values each piece at exactly $1/2$ because he divides the cake into two pieces he values equally. The second player receives the piece he prefers more among the two pieces, which therefore must be worth at least $1/2$ according to his valuation.

It is easy to see that in the cut and choose algorithm is that a player can obtain his fair share by following the algorithm, regardless of whether the other player also follows the algorithm or not.

It is also easy to see that in the case of two players proportionality implies envy-freeness.

3.2 Dubins-Spanier

The second cake-cutting algorithm which I discuss involves fairly dividing a cake amongst n players. This algorithm proposed in 1961 by Dubins and Spanier guarantees proportional allocations.

We observe that as we move from the two-player setting to the n-player setting, the algorithm gets much more complicated.

The algorithm proceeds as follows, as mentioned in [4]. An impartial referee moves a knife over the cake from left to right. When the knife reaches a point such that the piece of cake to the left of that point is worth exactly $1/n$ to one of the players, this player asks him to stop. The referee makes a cut at that point and the piece of cake to the left of the cut is given to that player. The player and allocated piece are then ignored in the rest of the process. The same process is repeated with the remaining players and the leftover cake until only one player is left. The last player receives the rest of the unclaimed cake.

This algorithm guarantees proportionality: The first $n-1$ players receive pieces they value at exactly $1/n$, hence the allocation for the first $n-1$ players is proportional. Clearly, the value of each of the first $n-1$ allocated pieces for the last player for is at most $1/n$. This is true because if it was greater than $1/n$ for any of those pieces, he would have asked the referee to stop and claimed the piece before the player who has currently claimed it. Therefore, the last player's value for the unclaimed piece is at least $1-(n-1)(1/n) = 1/n$ (using additivity). This implies that the allocation for the last player is also proportional.

As [4] mentions, the Dubins-Spanier algorithm can also be implemented through a discrete procedure, without the impartial referee. At each stage, each player makes a mark so the piece of cake to the left of the mark is worth $1/n$; this is where the player would stop the referee. We then allocate the piece of cake to the left of the leftmost mark to the player who made the mark. The same process is repeated with the remaining players and the leftover cake until only one player is left. The last player receives the rest of the unclaimed cake. Hence we simulate the previous algorithm in a discrete manner, and the same proof for proportionality goes through.

Unfortunately, the Dubins-Spanier algorithm does not guarantee envy-free allocations. Players do not envy other players allocated earlier because they have a value of at least $1/n$ for their piece and if their valuation for a piece allocated before was greater than $1/n$, they would have asked the referee to stop and claimed that piece before the player who has currently claimed it. However, players could envy players allocated later.

For example, as in [4], take a case of cake cutting where there are three players. Player 1 said to stop first and player 2 second, and player 3 received the remaining piece. Player 2 values his own piece at exactly $1/3$, has value of at most $1/3$ for the piece of player 1, but may have value as great as $2/3$ for the piece of player 3. This is clearly not an envy-free allocation as player 2 may envy player 3.

3.3 Even-Paz

I now discuss an algorithm that also guarantees proportional allocations for n players, but is much more efficient than the Dubins-Spanier algorithm. The algorithm proceeds as follows, as mentioned in [4].

For simplicity, we assume the number of players is a power of 2. Each time the procedure is executed, the players make marks where the piece to the left of the mark (and right of the mark) is valued at $1/2$. We now separate the players into two subsets of equal size, such that all marks made by the players of the first subset lie to the left (weakly) of the marks made by the players of the second subset. The players in the first subset then receive the piece of cake to the left of their rightmost mark, while the players in the second subset receive the remaining cake. The same procedure is applied recursively to the two subsets of players and the two corresponding pieces of cake, until each player has a single piece.

Each time the procedure is called, each player receives at least half of the remaining cake according to their value function. It is easy to see that each player participates in exactly $\log n$ calls to the procedure, because each call doubles the number of pieces and we need to reach n pieces. There the number of calls in exactly $\log n$. Each player therefore receives a piece of cake worth at least $(1/2)^{\log n} = 1/n$. Therefore, we see that proportionality is guaranteed. However, it is easy to see that envy-freeness is not guaranteed.

It is easy to see that the Even-Paz algorithm requires making fewer marks than the Dubins-Spanier algorithm. I will make this more precise in a future section where I discuss the complexity of cake cutting.

3.4 Selfridge-Conway

Selfridge and Conway in around 1960 designed an algorithm that guarantees envy-free allocations for three players.

The algorithm proceeds as follows, as mentioned in [4].

Stage 1 (Partitioning into Cake 1 and Cake 2): Player 1 divides the cake into three equal pieces according to his valuation. Player 2 trims the largest piece, such that the two largest pieces are of equal value according to his valuation function. We call the cake without the trimmings Cake 1 and the trimmings Cake 2.

Stage 2 (division of Cake 1): Player 3 chooses the largest piece of Cake 1 according to his valuation. If player 3 did not choose the trimmed piece, player 2 is allocated the trimmed piece. Otherwise, player 2 chooses the larger of the two remaining pieces. Either player 2 or player 3 receives the trimmed piece; we denote that player by T and the other player by T'. Player 1 is allocated the remaining untrimmed piece.

Stage 3 (division of Cake 2): T' divides Cake 2 into three equal pieces according to his valuation. Players T, 1, and T' choose the pieces of Cake 2, in that order.

The division of Cake 1 is envy-free: Player 3 chooses first, and therefore does not envy anyone. Player 2 likes the trimmed piece and another piece equally and is guaranteed to receive one of these two pieces. Player 1 is indifferent between the two untrimmed pieces and does indeed receives an untrimmed piece.

Now we divide Cake 2. Player T goes first and hence does not envy the others; and $T^{'}$ is indifferent amongst the three pieces of Cake 2. Player 1 does not envy $T^{'}$ but may prefer the piece of Cake 2 allocated to T to his own piece of Cake 2. However, that does not matter because even if we allocated all of Cake 2 to T, we would reconstruct just one of the original three pieces cut by player 1 (worth $1/3$ to him); but player 1 already received a piece worth 1/3 at the end of stage 1.

Thus, we conclude that the above algorithm guarantees envy-free allocations. Recall that envy-freeness implies proportionality, hence the above algorithm also guarantees proportional allocations.

3.5 Complexity

To analyse the complexity of the above algorithms formally we must adopt a precise model that specifies which operations/queries a cake cutting algorithm is allowed to use. Complexity will then be measured by counting/bounding the number of required queries.

The standard concrete complexity model for cake cutting is the Robertson-Webb model mentioned in [5], introduced by Robertson and Webb in 1998. The model allows the following two operations/queries.

- $\text{eval}_i(x, y)$: Asks agent i to evaluate the interval $[x, y]$. Formally, $\text{eval}_i(x, y) = V_i(x, y)$.
- $\text{cut}_i(x, \alpha)$: Asks agent i to cut a piece of cake worth a given value α, starting at a given point x. Formally, $\text{cut}_i(x, \alpha) = y$ where y is the leftmost point such that $V_i(x, y) = \alpha$.

The Robertson-Webb model seems to be very simple and restrictive, but is able to simulate the cake cutting algorithms described until now in the paper.

For example, we can easily simulate the cut and choose algorithm using the Robertson-Webb model. As mentioned in [5], the algorithm sends a $\text{cut}_1(0, 1/2)$ query to player 1. Player 1 returns a point y in response such that $V_1(0, y) = V_1(y, 1) = 1/2$. We now send agent 2 an $\text{eval}_2(0, y)$ query. If the answer is at least 1/2, we know that $A_1 = [y, 1]$, $A_2 = [0, y]$ is a proportional allocation; if not we can obtain a proportional allocation by switching the two pieces.

Now I can formally show that the Even-Paz algorithm is more computationally efficient than the Dubins-Spanier algorithm according to the Robertson-Webb model.

The Dubins-Spanier algorithm can be simulated by sending each remaining player a $\text{cut}_i(x, 1/n)$ query, where x is the left boundary of the remaining cake. The total number of queries is therefore $1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$. The overall number of queries is $\Theta(n^2)$.

The Even-Paz algorithm requires sending a $\text{cut}_i(y, V_i(y, z)/2)$ query to each player in each recursive call, where $[y, z]$ is the current piece. Assuming that n is a power of 2, it is easy to see that each player participates in exactly $\log n$ calls, because each call doubles the number of pieces and we need to reach n pieces. The overall number of queries is therefore $\Theta(n \log n)$, which is much smaller than $\Theta(n^2)$.

4 Further Work

4.1 Recent Progress

The Selfridge-Conway algorithm is a great solution for the three-player case of envy-free cake cutting. However, an algorithm for the n-player case would have to wait more than three decades.

In 1992, Brams and Taylor came up with the first envy-free cake-cutting algorithm for an arbitrary number of players.

Unfortunately, their algorithm has a major flaw: it has unbounded running time.

It had been a major open problem whether there exists a discrete and time-bounded envy-free cake cutting algorithm for any number of players. In 2016, Aziz and Mackenzie [2] resolved the problem by proposing a discrete and bounded envy-free protocol for any number of players

However, their algorithm has a major flaw as well: the number of queries required by the algorithm is $\mathcal{O}(n^{n^{n^{n^{n^{n}}}}})$, which is definitely not in P.

4.2 Variants of the setup

In the envy-free cake cutting we spoke about until now, the cake must be divided amongst players in a way such that each agent weakly prefers their own piece over the piece of *every* other player. This is known as global envy-freeness.

We can consider a variant of this notion of global envy-freeness, in which there is some specified graph on the players(each player is a node), and players only evaluate their pieces relative to their neighbors' in the graph. This is a local view, whereas before we were taking a global view.

Specifically, as [1] mentions, we say that an allocation is locally envy-free if no player envies a neighbor's piece(a player envies his neighbor if he prefers the neighbor's piece over his own piece) and is locally proportional if each player values his own piece at least as much as the average of the values of his neighbor's pieces.

It is easy to see that local envy freeness implies local proportionality and global envy-freeness implies local envy-freeness. However, global proportionality does not imply local proportionaity, or vice versa.

I worked on finding efficient locally envy-free free algorithms for special graphs which run faster than the globally envy-free algorithm, which hopefully in polynomial time.

I did make some progress on a variant of the above problem. I considered a special case of cake cutting where I am allowed to have a wasteful allocation where some part of the cake may be un-allocated. Let us say I am allowed to waste ϵ fraction of the cake according to some players value function. Let us call this ϵ-wasting.

Initially, I came up with locally envy free algorithms for path graphs with 5 or less vertices for the ϵ-wasting case. See [8] for the definition of path graphs.

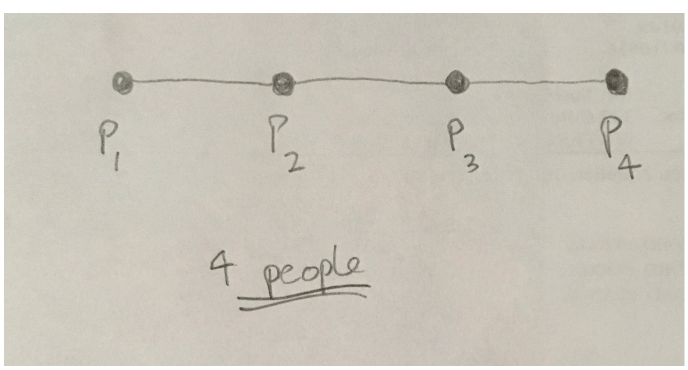
I recently extended the above algorithm to path graphs of arbitrary number of vertices. The algorithm runs in time polynomial in n and $1/\epsilon$.

5 Proposed Algorithm

In this section I discuss an algorithm for local envy freeness in path graphs of any length. However, this algorithm is ϵ-wasting.

5.1 4 agents

For ease of explanation, I initially start with a small example. I first discuss an ϵ-wasting algorithm for locally envy free cake cutting in path graphs of length 4. Consider the line graph of 4 players shown below.



I start from the rightmost player P_4. P_4 cuts the cake into 4 equal parts according to his valuation.

P_3 now chooses the top 3 pieces according to his valuation and P_4 gets the remaining piece.

P_3 now trims the two larger pieces to make it equal to the smallest piece according to his valuation. The trimmings are kept aside.

Now P_2 picks the top two pieces according to his valuation and P_3 gets the remaining piece.

Now P_2 trims the larger piece to make it equal to the smaller piece according to his valuation. The trimmings are kept aside.

P_1 chooses the larger piece and P_2 gets the remaining piece.

Now we can collect the trimmings and repeat the same process again multiple times starting from P_4. The trimmings will keep reducing in size and will eventually become arbitrarily small.

It is easy to see that the algorithm is locally envy free.

P_4 is not envious of P_3 because he split the cake into 4 equal pieces and got a piece of value $1/4$. The value of each of the other pieces is $1/4$ and P_3 can either get a piece of value $1/4$(if he gets an untrimmed piece) or a piece of value $\leq 1/4$ (if he gets a trimmed piece)

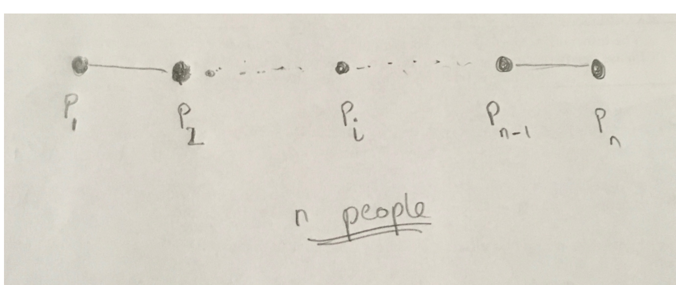
Using similar reasoning P_3, P_2 are not envious of P_2, P_1 respectively.

P_3 is not envious of P_4 because once P_4 cuts the cake into 4 equal parts according to his valuation, P_3 chooses the top 3 pieces and P_4 gets the remaining piece. Therefore, P_3 even weakly prefers the smallest piece amongst the 3 he chose over the piece of P_4 and he eventually after the trimmings does get a piece of the same value as the smallest piece and weakly prefers this piece over the piece of P_4.

Using similar reasoning P_1, P_2 are not envious of P_2, P_3 respectively.

5.2 Any number of agents

I now generalize the above algorithm and discuss an ϵ-wasting algorithm for local envy freeness in path graphs of length n. Consider the below line graph of n players.



I start from the rightmost player P_n. P_n cuts the cake into n equal parts according to his valuation.

P_{n-1} now chooses the top $n-1$ pieces according to his valuation and P_n gets the remaining piece.

P_{n-1} now trims the $n-1$ larger pieces to make it equal to the smallest piece according to his valuation. The trimmings are kept aside.

Now P_{n-2} does almost the same thing that P_{n-1} did.

In general, when P_i cuts the cake into i equal parts according to his valuation, P_{i-1} chooses the top $i-1$ pieces from these according to his valuation and P_i gets the remaining piece.

P_{i-1} now trims the $i-2$ larger pieces to make it equal to the smallest piece. The trimmings are kept aside.

The players keep follow this process until one piece reaches P_1.

Now we can collect the trimmings and repeat the same cycle again multiple times starting from P_n, and the trimmings will keep reducing in size and will eventually become arbitrarily small.

It is easy to see using a similar reasoning as in the case of 4 people that the algorithm is locally envy free.

For every $2 \leq i$, P_i is not envious of P_{i-1} because he split the cake into i equal pieces according to his valuation and got a piece of value $1/i$. The value of each of the other pieces is $1/i$ and P_{i-1} can either get a piece of value $1/i$ (if he gets an untrimmed piece) or a piece of value $\leq 1/i$ (if he gets a trimmed piece).

For every $i \leq n-1$, P_{i-1} is not envious of P_i because once P_i cuts the cake into i equal parts according to his valuation, P_{i-1} chooses the top $i-1$ pieces amongst according to his valuation and P_i gets the remaining piece. He even weakly prefers the smallest piece amongst the $i-1$ he has over the piece of P_i and he eventually after the trimmings does get a piece of the same value as the smallest piece and weakly prefers this piece over the piece of P_i.

5.3 Complexity Analysis

It is easy to see that each cycle takes $O(n^2)$ time. P_n is queried n times. In general P_i is queried i times. The total number of queries in one cycle is therefore $1+2+3+\ldots+n = \frac{n(n+1)}{2} \in O(n^2)$.

Now we need to calculate the number of cycles we need to do so that the trimmings has values lesser than ϵ according to P_n. Total running time is $O(kn^2)$, if the number of cycles is k.

After one cycle the trimmings have value at most $\frac{n-1}{n}$ because after the cycle P_n has a piece of value $\frac{1}{n}$ according to his valuation. It is easy to see that after k cycles the trimmings will have value bounded by $(\frac{n-1}{n})^k$.

We need to find k such that

$$(\frac{n-1}{n})^k = \epsilon$$

Taking $\log$ on both sides and after manipulation

$$k = \frac{\log(1/\epsilon)}{\log(\frac{n}{n-1})}$$

Now, we know that

$$\begin{aligned}
&\log x > 1 - \frac{1}{x} \text{ for } x > 1 \\
\implies &\log \frac{n}{n-1} > 1 - \frac{n-1}{n} \\
\implies &\log \frac{n}{n-1} > \frac{1}{n} \\
\implies &k < n \log \frac{1}{\epsilon}
\end{aligned}$$

Total running time is $O(kn^2) = O(n^3 \log(\frac{1}{\epsilon}))$. The algorithm runs in time polynomial in n and $1/\epsilon$

6 Conclusion and Future Work

The paper discusses a variant of envy-free cake cutting on graphs (called local envy-freeness) where envy-freeness is only required with respect to an agent's neighbors.

I concentrate on a variant of the standard setup where I am allowed to have a wasteful allocation where some part of the cake may be un-allocated (ϵ-wasting).

I give a locally envy free algorithm for path graphs with any number of vertices for the ϵ-wasting case. The algorithm runs in time polynomial in n and $1/\epsilon$.

For future work, I hope to come up with efficient locally envy free algorithms for the ϵ-wasting case for special graphs apart from the path graph.

[3] provides some good directions for future work as well.

Also, as one of my reviewers suggested, I've only looked at two ways of making the envy-free cake cutting problem tractable, which are restricting the graph structure(local envy-freeness) and allowing wasteful allocations(ϵ-wasting). I could also look at other variants where the valuation functions of each player are further restricted (such as piecewise constant) or where the allocations are restricted (such as requiring that each player can only get a contiguous piece).

References

[1] Rediet Abebe, Jon Kleinberg, and David C Parkes. Fair division via social comparison. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 281–289. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[2] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 416–427. IEEE, 2016.

[3] Xiaohui Bei, Youming Qiao, and Shengyu Zhang. Networked fairness in cake cutting. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3632–3638, 2017.

[4] Ariel D. Procaccia. Cake cutting: Not just child's play. *Commun. ACM*, 56(7):78–87, July 2013.

[5] Ariel D Procaccia. Cake cutting algorithms. 2014.

[6] Wikipedia contributors. Fair cake-cutting — Wikipedia, the free encyclopedia, 2018. [Online; accessed 16-August-2018].

[7] Wikipedia contributors. Fair division — Wikipedia, the free encyclopedia, 2018. [Online; accessed 16-August-2018].

[8] Wikipedia contributors. Path graph — Wikipedia, the free encyclopedia, 2018. [Online; accessed 16-August-2018].