On Fair Allocation of Mixed Goods

TDT4501 - Computer Science, Specialization Project

Sivert Utne

December 9, 2022

Preface

This report is written as part of the course *TDT4501 - Computer Science, Specialization Project*. The intended purpose of this report is a pre-study for a master thesis.

Thank you to the following people for their help, support and guidance:

Supervisor: *Magnus Lie Hetland*

Supervisor: *Halvard Hummel*

Abstract

In the world of fair allocation of indivisible items and fair division of divisible items it might be wise to look at the combination of these two field. This will be the fair allocation and division of mixed goods. There already exists some work in the relevant literature about this topic, this report aims to summarize and compare these works and algorithms by looking at the allocation they each give. Finally these results will be compared to several naive/simpler algorithms to see which situations suits the different algorithms best, and if they are necessary.

Contribution

This report will aim to give an overview over the present literature on fair allocation and fair division in a mixed goods setting. The report compares previous algorithms against eachother and several naive solutions to the problem. The report concludes with a overview over which situations the different algorithms are suited.

Contents

Preface **i**

Abstract **ii**

Contribution **iii**

1 Introduction **1**

1.1 Preliminaries 2

1.2 Previous Work 5

1.2.1 Mixed Goods 5

1.2.2 Mixed Resources 5

2 Method **6**

2.1 Previous Solutions 7

2.2 Our Solutions 8

2.2.1 Frozen Cake 8

2.2.2 Frozen Cut Cake 8

3 Results **12**

3.1 Discussion 19

4 Conclusions **20**

4.1 Future Work 21

Bibliography **22**

List of Figures

2.1 Visualization of agen's valuations in an instance, these valuations are used in the following figures. 9

2.2 Visualization of each agents MMS Allocation for the mixed instance. 10

2.3 Visualization of each agents MMS Allocation for the mixed instance. 10

2.4 Visualization of each agents MMS Allocation for the mixed instance. 11

2.5 Visualization of each agents MMS Allocation for the mixed instance. 11

3.1 Regular indivisible Algorithm vs Mixed Algorithm. 12

3.2 Regular indivisible Algorithm vs Mixed Algorithm. 13

3.3 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Small Cakes. 13

3.4 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Small Cakes. 14

3.5 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Medium Cakes. 14

3.6 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Medium Cakes. 15

3.7 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Large Cakes. 15

3.8 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Large Cakes. 16

3.9 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Variable Cakes. 16

3.10 Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Variable Cakes. 17

3.11 Approximate vs. Exact MMS calculation for the Mixed Algorithm. 17

3.12 Approximate vs. Exact MMS calculation for the Mixed Algorithm. 18

List of Tables

3.1 Running time for the three algorithms. 18

List of Algorithms

1 Algorithm for Frozen Cake 8

Chapter 1

Introduction

This report will look at various approaches and algorithms proposed in the relevant literature for how to handle a mixed goods setting. In Section 1.1 we will define the problem and the terminology used in the report. In Section 1.2 we will look at the relevant literature and the different approaches that have been proposed. In Chapter 2 we will present the approach used to compare and analyze these different approaches. In Chapter 3 we will present the experiments we have conducted and the results we have obtained. Finally, in Chapter 4 we will conclude the report and discuss future work.

1.1 Preliminaries

Collection of terminology, notation and definitions that are presumed to be known for the rest of the report.

Mixed Goods

When discussion mixed goods we are talking about a mix of both indivisible (i.e. paintings, cars, equipment etc.) and divisible goods (often referred to as *cake*). A mixed goods instance can contain any combination of number of divisible and indivisible goods. A good means that all items will have a positive utility or value for all agents.

Problem Instance

An instance $\langle N, M, V, F\rangle$ of the fair allocation of mixed goods problem is defined by a:

- set N of $n \in N$ agents
- set M of $m \in M$ indivisible items
- set V of $v \in V$ valuation functions where v_i is valuation function for agent i
- set F of $f \in F$ density functions where f_d divides divisible item d

Additive valuations

A well-studied subclass of monotone valuations is that of additive valuations, wherein an agent's value of any subset of items is equal to the sum of the values of individual items in the set, i.e., for any agent $i \in N$ and any set of items $S \subseteq M$, $v_i(S) := j \in Svi(j)$, where we assume that $v_i(\emptyset) = 0$. For simplicity, we will write $v_i(j)$ or $v_{i,j}$ to denote $v_i(j)$.

Bundle

The collection of items that an agent receives is called a bundle. A bundle is a subset of the items in M.

Allocation

An allocation $A := (A1, ..., An)$ is an n-partition of a subset of the set of items M, where $Ai \subseteq M$ is the *bundle* allocated to the agent i (note that A_i can be empty $\emptyset$). An allocation is said to be complete if it assigns all items in M, and is called partial otherwise.

Top-trading envy graph

The top-trading envy graph T_A of an allocation A is a subgraph of its envy graph G_A with a directed edge from agent i to agent k if $v_i(A_k) = max_{j \in N} v_i(A_j)$ and $v_i(A_k) > v_i(A_i)$, i.e., if agent i envies agent k and A_k is the most preferred bundle for agent i.

Envy Graph

The envy graph GA of an allocation A is a directed graph on the vertex set N with a directed edge from agent i to agent k if $vi(Ak) > vi(Ai)$, i.e., if agent i prefers the bundle Akover the bundle Ai.

Envy-freeness and its relaxations

EF
An allocation A is said to be envy-free (EF) if for every pair of agents $i, k \in N$, we have $v_i(A_i) \geq v_i(A_k)$. In other words, no agent prefers the bundle of another agent over their own bundle.

EF1
An allocation A is said to be envy-free up to one item (EF1) if for every pair of agents $i, k \in N$ such that $A_i \cap A_k \backslash = \emptyset$, there exists an item $j \in A_i \cup A_k$ such that $v_i(A_i \backslash j) \geq v_i(A_k \backslash j)$.

Envy-freeness for mixed goods (EFM)
An allocation A is said to satisfy envy-freeness for mixed goods (EFM) if for any agents $i, j \in N$,

- if agent j's bundle consists of only indivisible goods, there exists $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \backslash g)$;

- otherwise, $u_i(A_i) \geq u_i(A_j)$.

It is easy to see that when the goods are all divisible, EFM reduces to EF; when goods are all indivisible, EFM reduces to EF1. Therefore EFM is a natural generalization of both EF and EF1 to the mixed goods setting.

1.2 Previous Work

1.2.1 Mixed Goods

The first article we will look at is

[3] and [2] [5]

1.2.2 Mixed Resources

[4]

Chapter 2

Method

The method we will use to analyze and compare the different algorithms will be as follows.

We will create a large collection of randomly created instances/situations. Each of the algorithms will then be tasked with dividing the items. These allocations and division will then be compared directly in regards to the average score each agent gives their own bundle, and the total valuation for all agent combined. Because all algorithms will use the exact same items with the exact same valuations and agents, a comparison where one algorithms clearly outperforms the others wil be a clear sign that this algorithm is superior for that specific situation. The instances will be divided into a number of groups:

- **Single Small Cake:** Instance with at least 1 indivisible item and 1 small cake (the indivisible items are dominant).
- **Single Large Cake:** Instance with at least 1 indivisible item and 1 large cake (the cake is dominant).
- **Multiple Cakes:** Instances with multiple divisible items.

in addition to these metrics the algorithms will also be compared in regards to the time it takes to run the algorithm, and the general complexity/robustness the algorithm has. A very simple algorithm that perform just as well as, or close to, a complex algorithm can be considered a better algorithm. These evaluations are done in Chapter 3 and finally presented in Chapter 4.

2.1 Previous Solutions

2.2 Our Solutions

2.2.1 Frozen Cake

A simplification of the problem can be made by assuming the cake is *frozen*, i.e. The cake is no longer divisible and must be allocated along with the indivisible items. This simplification naturally leads to an EF1 allocation with additive valuations. This does however remove the EFM guarantee as with a large cake, an agent with cake can now be envied, contradicting the rule of EFM.

The algorithm for this is a direct copy of the algorithm desscribed in [1]. Which is shown in Algorithm 1, pay attention to line 3-7.

Algorithm 1 Algorithm for Frozen Cake

Require: n number of items, m number of resources, c_i cost of item i, r_{ij} resource j required by item i, R_j resource j available
Ensure: $n \geq 0$

```
1:  X ← x
2:  N ← n
3:  while N ≠ 0 do
4:      if N is even then
5:          X ← X × X
6:          N ← N/2                          ▷ This is a comment
7:      else if N is odd then
8:          y ← y × X
9:          N ← N − 1
10:     end if
11: end while
```

2.2.2 Frozen Cut Cake

A variation of the simplification in Section 2.2.1 will be to assume the cake is cut into n equally sized pieces. Again this simplification naturally leads to an EF1 allocation with additive valuations.

We will further investigate how many pieces a cake should be cut in. A cake can potentially be cut into ∞ pieces, which will return the problem to the original problem. We are here instance focusing on a cake cut into a number of pieces n that grows proprotinally with the number of agents.

Again we use Algorithm 1, with the only difference being that the cake is cut into n pieces. We will repeat the experiments with $\frac{n}{2}$, $2n$ and $4n$ pieces of cut cake.

Figure 2.1: Visualization of agen's valuations in an instance, these valuations are used in the following figures.

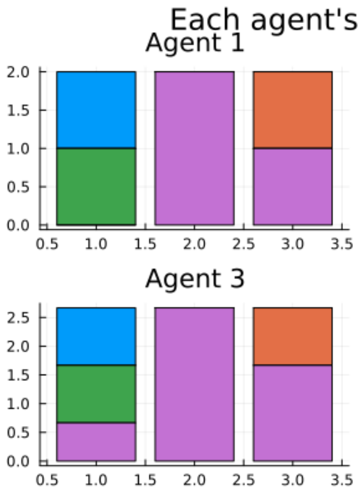

Figure 2.2: Visualization of each agents MMS Allocation for the mixed instance.

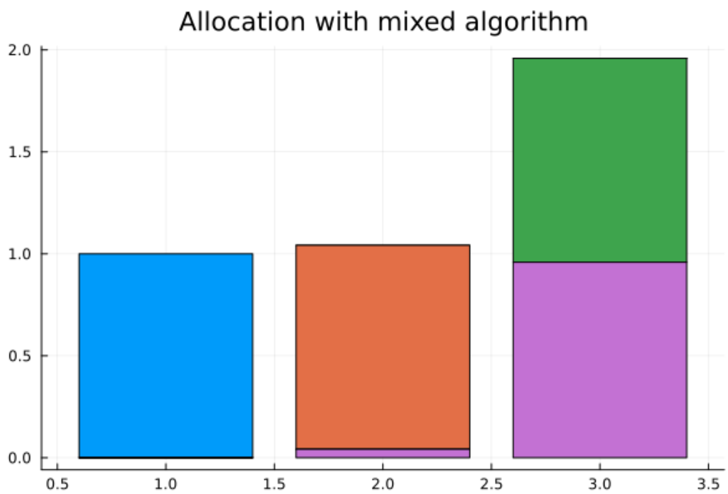

Figure 2.3: Visualization of each agents MMS Allocation for the mixed instance.

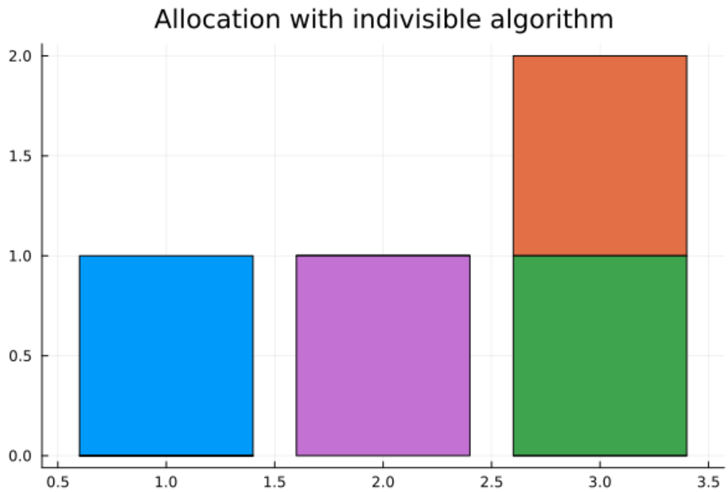

Figure 2.4: Visualization of each agents MMS Allocation for the mixed instance.

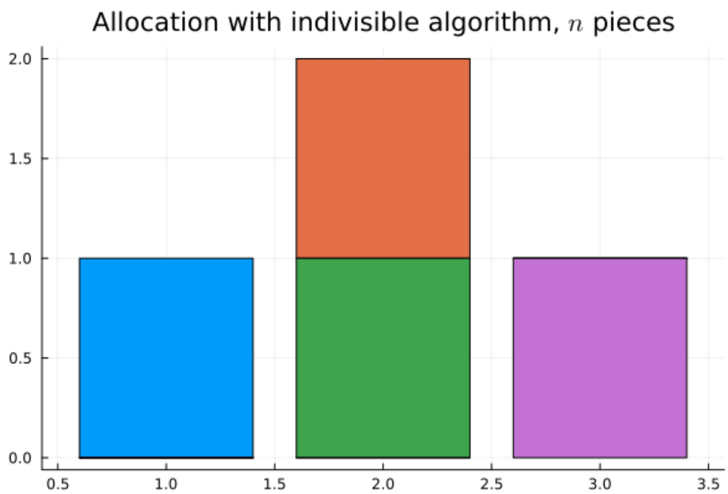

Figure 2.5: Visualization of each agents MMS Allocation for the mixed instance.

Chapter 3

Results

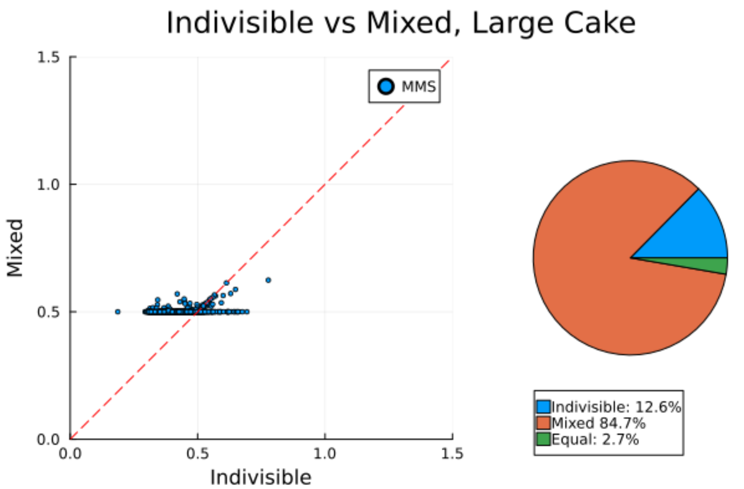

Figure 3.1: Regular indivisible Algorithm vs Mixed Algorithm.

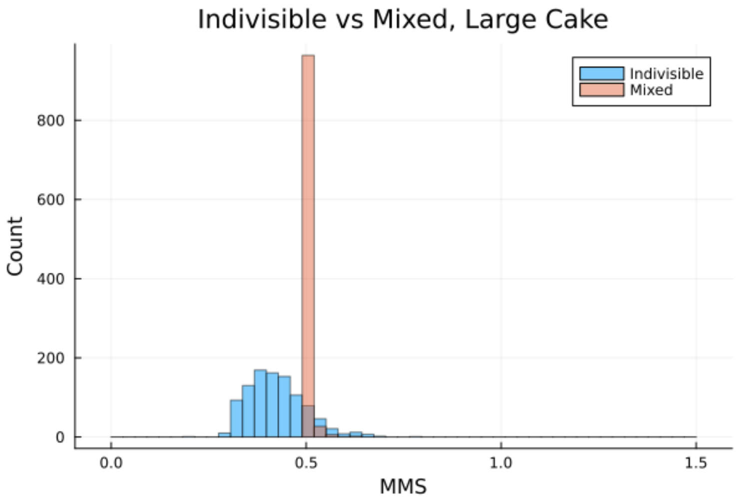

Figure 3.2: Regular indivisible Algorithm vs Mixed Algorithm.

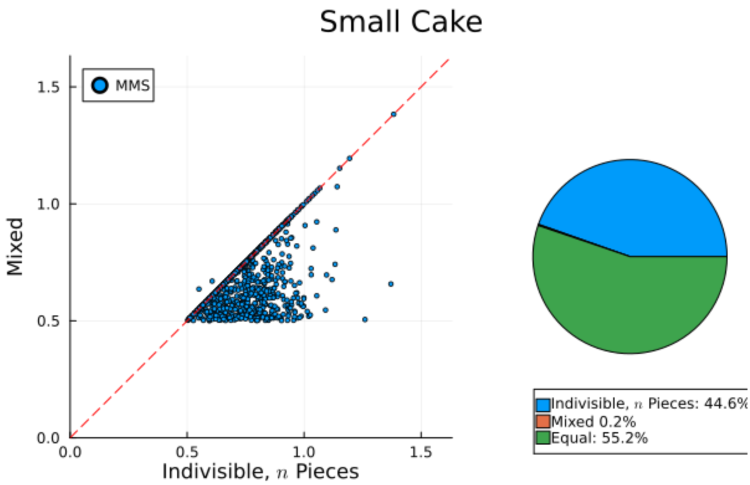

Figure 3.3: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Small Cakes.

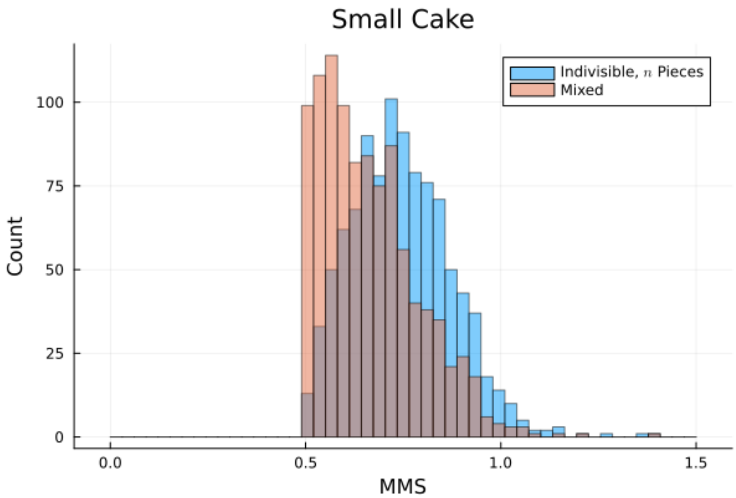

Figure 3.4: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Small Cakes.

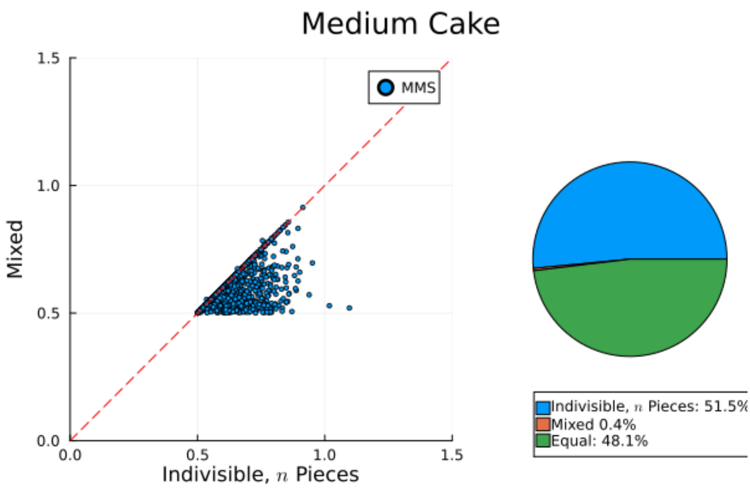

Figure 3.5: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Medium Cakes.

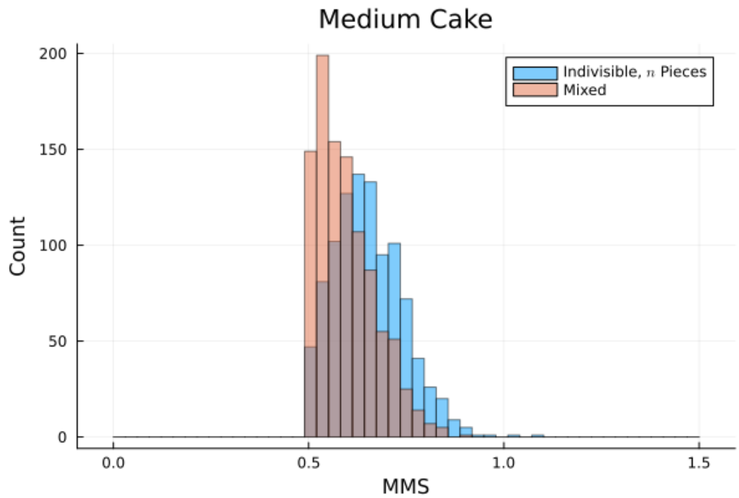

Figure 3.6: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Medium Cakes.

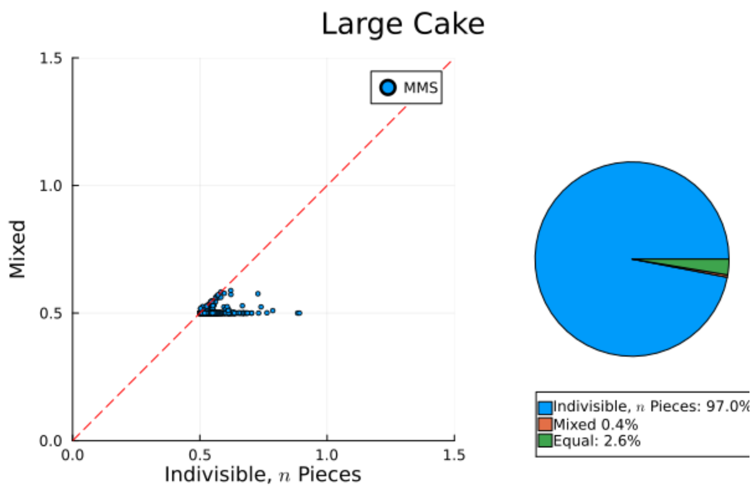

Figure 3.7: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Large Cakes.

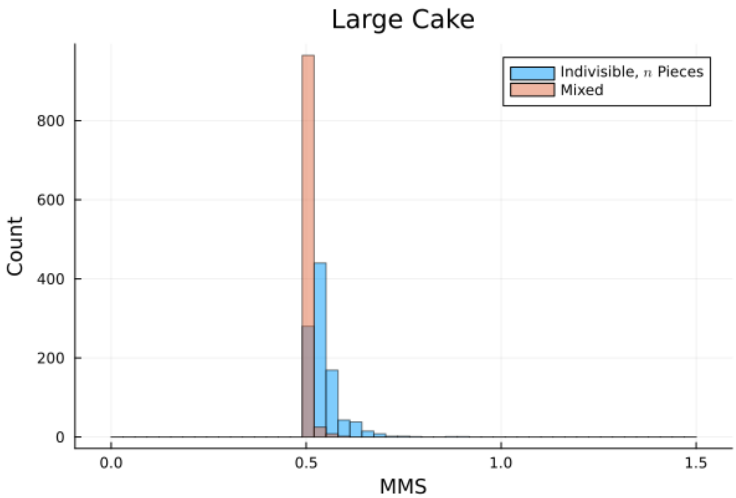

Figure 3.8: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Large Cakes.

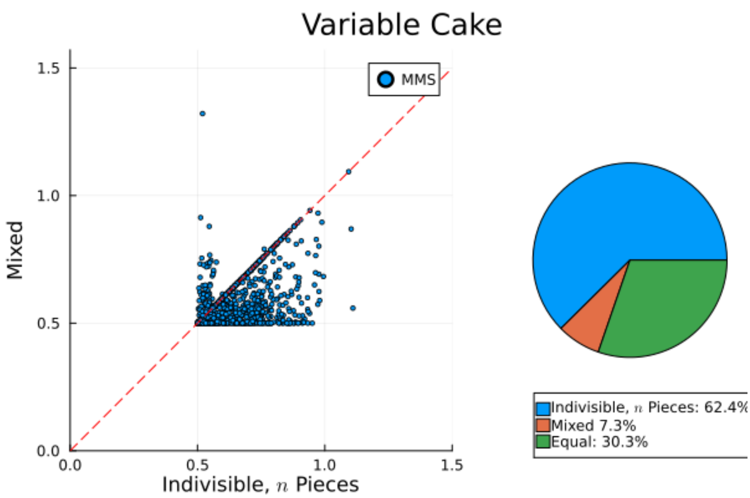

Figure 3.9: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Variable Cakes.

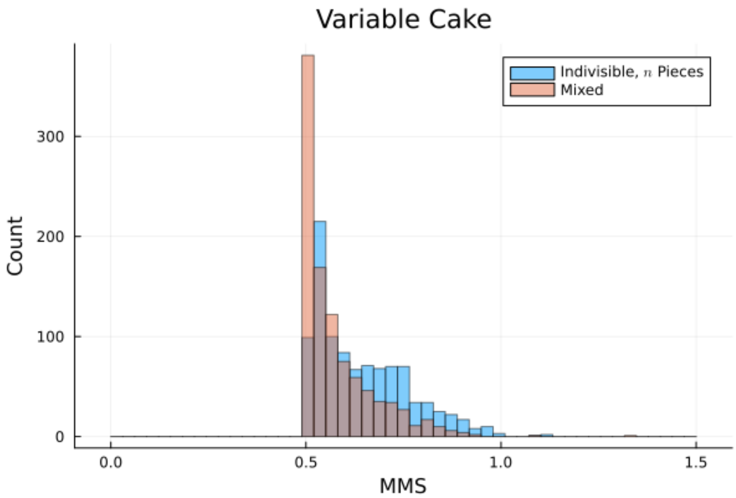

Figure 3.10: Indivisible Algorithm Cutting cake into n pieces vs. Mixed Algorithm for Variable Cakes.

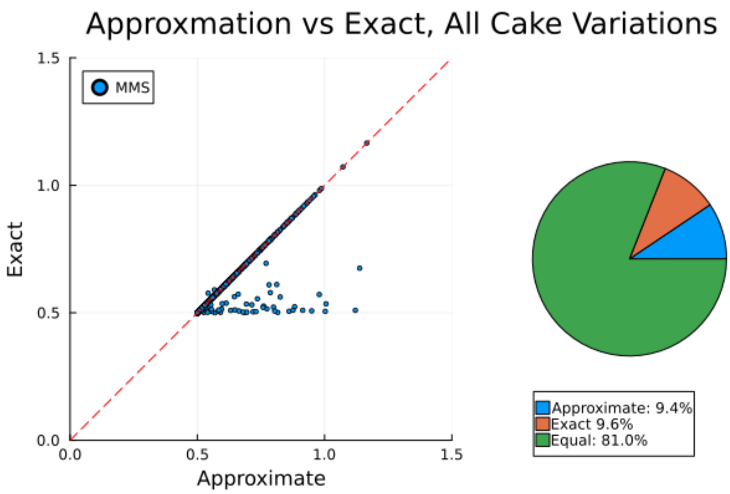

Figure 3.11: Approximate vs. Exact MMS calculation for the Mixed Algorithm.

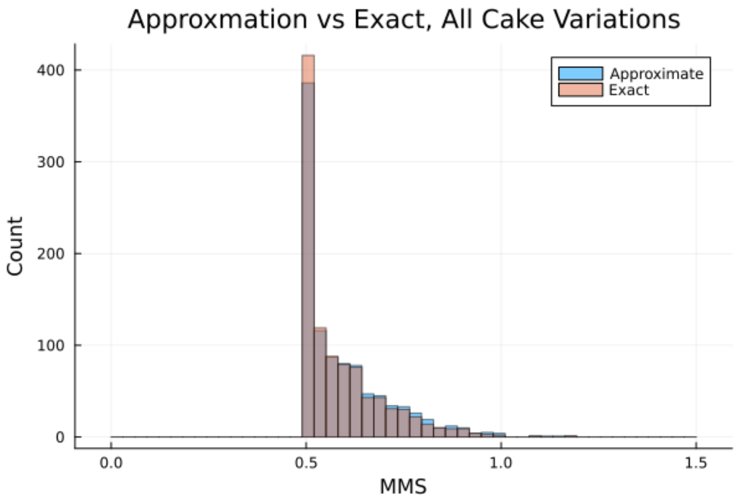

Figure 3.12: Approximate vs. Exact MMS calculation for the Mixed Algorithm.

Table 3.1: Running time for the three algorithms.

	Indivisible n Pieces	Mixed Exact	Mixed Approxmation
Small Cake	100.0ms	100.0ms	100.0ms
Medium Cake	100.0ms	100.0ms	100.0ms
Large Cake	100.0ms	100.0ms	100.0ms
Variable Cake	100.0ms	100.0ms	100.0ms

3.1 Discussion

Discussion of the results

Chapter 4

Conclusions

4.1 Future Work

Bibliography

[1] ? *?* ?, June 2021.

[2] Xiaohui Bei, Shengxin Liu, Xinhang Lu, Jinyan Liu, and Zihao Li. *Fair Division of Mixed Divisible and Indivisible Goods.* Unknown, January 2021.

[3] Xiaohui Bei, Shengxin Liu, Xinhang Lu, and Hongao Wang. *Maximin fairness with mixed divisible and indivisible goods.* Springer Science, June 2021.

[4] Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. *On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources.* Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

[5] Rucha Kulkarni, Ruta Mehta, and Setareh Taki. *Indivisible Mixed Manna: On the Computability of MMS+PO Allocations.* Unknown, April 2021.