Explorations in Envy-Free Allocations

Integer Optimization - Final Research Project
Professor Eric Friedman

Christopher Landgrebe, Calvin Suster, and Wyatt Walsh
May 8th, 2020

Notes to the Reader

The format and length of this report slightly deviate from the course project specifications. Due to the extended nature of the included analysis and accompanying visualizations, the authors suggest to utilize the link found below to access the accompanying cloud-hosted interactive notebook for use while reading. This notebook contains interactive plots as well as variables for the different results datasets. Please note that this service typically takes a couple of minutes to initialize. Furthermore, the entire project repository can be accessed by clicking "visit this repo" within the notebook environment, or by using the second link below.

| Virtual Project Notebook Direct Link | Project Repository Direct Link |

Contents

1 Executive Summary

Fair division problems are a significant class of problems with considerable multidisciplinary involvement ranging from social science to computer science. These problems first emerged shortly after WWII, with the first known problem's goal being to find ways to cut cake such that the cake is fairly allocated [2, 13]. Since then, a myriad of different problems, fairness definitions, and implementations have been created in an attempt to aptly model a diverse set of situations. This body of work explores a particular fairness criterion known as envy-freeness.

Currently there exist many specificies of envy-freeness, applied to a multitude of scenarios, solved through assorted methodologies [1]. To guide the work in this project, three particular definitions of envy-freeness are analyzed for a particular situation. These are envy-freeness, envy-freeness up to one item, and envy-freeness with the inclusion of a divisible subsidy in the form of a cash amount. We apply these definitions to the situation where items are indivisible and and valuations are both additive and normalized. To solve these problems, three mixed integer program formulations are created. These formulations are then modeled in the *AMPL* programming language and solved using the *IBM CPLEX* solver for two simple examples and a collection of generated data for different combinations of number of people and number of items to be allocated. The results for the two simple examples serve to validate the accuracy of the formulations and the results for the collection of generated data allow for analysis to be conducted on the complexity of these problem types. Furthermore, strategies are devised and implemented to reduce the runtime of the envy-freeness instance including: upper-bounding the objective function, initializing *CPLEX* with a feasible starting solution, the combination of both upper-bounding the objective function and initializing *CPLEX* with a feasible starting solution, and finally tuning various *CPLEX* parameters.

The results for all three fairness criteria as well as the four runtime improvement strategies are analyzed through the use of a multitude of visualization techniques. A few takeaways include that the total subsidy value serves as a scaling parameter to the envy-freeness as well as the fact that the theory presented by Caragiannis et al. [5] proving the existence of envy-freeness up to one item is reflected in this work. Most notably, however, the patterns present in the generated visualizations support the discussions of Dickerson et al. (2014) [8], in addition to Manurangsi and Suksompong [11, 12]. In particular, the research by Manurangsi and Suksompong (2019) into the number of people and number of items needed to satisfy the existence of an envy-free allocation, as well as the phase transition from non-existence to existence are highly approximately viewable in the visualizations created [12]. The visualizations of solver elapsed time for a certain combination of number of people and number of items reflect exactly the notion presented by Dickerson et al. (2014) as the value of solver time rises up to the equality of number of people, number of items and then is reflected and falls [8]. By coupling the visualizations of solver elapsed time for a certain combination of number of people and number of items with the visualizations of the approximate envy-freeness value an understanding of the complicated relationship between envy-freeness and complexity begins to be unearthed.

2 Technical Report

2.1 Preliminaries

We begin by introducing necessary notation and mathematical definitions in preparation for further elaboration of technicalities. Throughout this paper, the set of m number of people is defined as $P = \{1, 2, ..., m\}$ and the set of n number of items is defined as $I = \{1, 2, ..., n\}$. A number x is said to be indivisible if $\nexists\, y \mid \exists \frac{x}{y} \forall y$ where y is also a number. A bundle, B_k, is the subset of items in I allocated to person k. $v_{i,j}$ is value of item j to person i. Person i's bundle's value to person j is written as $\mathrm{v}_j(B_i)$. For the generated datasets, $v_{i,j} \in U[0, 1]\ \forall\ i, j$. These valuations are additive such that $\sum_{j \in B_k} v_{i,j} = \mathrm{v}_i(B_k)$. Prior to evoking any solver instances, except for the case of p-envy-free with a subsidy, values are normalized as $v_{i,j} = \frac{v_{i,j}}{\sum_{j \in I} v_{i,j}}\ \forall\ i \in P$. An allocation, A, is considered to be a partitioning of I into m bundles with B_k being allocated to person k such that: $\cup_{i \in P} B_k = I$, and $B_k \cap B_j = \varnothing\ \forall\ k \neq j$.

AMPL was used to create the data files for these generated examples since it simplifies the process of creating .dat files for use in .run scripts. *AMPL*'s built-in *Uniform01* function is used to create random item values of which then populate matrices of size m by n for the corresponding files representing m people and n items. The resultant matrices are output, along with the sets of people and items to a .dat file for all combinations of people and items from one to twenty of each. The full implementation of this data generation can be found in the file "data.run" and the resultant .dat files are labeled "1" through "400", both found in the repository. In the case of p-envy-freeness with a subsidy, a subset of these files representing combinations of number of people and number of items from one to fifteen is taken and relabeled "1" through "225" in a subdirectory.

2.2 p-Envy-Freeness (pEF)

2.2.1 Explanation of Problem Definition

The first problem considered is p-envy-freeness of indivisible items amongst a group of people with additive and normalized valuations. All items are required to be allocated in order to avoid the trivial envy-free allocation that could exist if all items were disposed. Here, p-envy-freeness is considered to be the largest difference between how a single person values another person's bundle and how that same person values their own bundle amongst all possible pairs of people of that instance. Item indivisibility is an important feature for this section as it implies that many times completely envy-free ($p = 0$) allocations may not exist and introduces binary variables into the program, which in turn greatly increase the complexity. On the other hand, the additive and normalized aspects of the personal valuations of items lends an aspect of simplicity and lack of realism to the problem. Leveraging the work done by Lipton et al., a formal definition of the pairwise maximum envy for a certain allocation is [9]:

$$p = \max_{i,j} \left(\mathrm{v}_i(B_j) - \mathrm{v}_i(B_i)\right) \forall\ i, j \in P \mid i \neq j \tag{1}$$

Thus, the overall objective of the problem is:

$$\min p \tag{2}$$

2.2.2 Mixed Integer Program Formulation and Explanation

$$\begin{aligned}
\text{minimize} \quad & p && && (1)\\
\text{subject to} \quad & \sum_{i \in P} x_{i,j} = 1 && \forall\, j \in I && (2)\\
& \sum_{j \in I} \left(v_{i,j} x_{k,j} - v_{i,j} x_{i,j}\right) - p \leq 0 && \forall\, i,k \in P : i \neq k && (3)\\
& p - 1 \leq 0 && && (4)\\
& p \geq 0 && && (5)\\
& x_{i,j} \in \{0,1\} && && (6)
\end{aligned}$$

The decision variables for this problem are $x_{i,j}$, binary (0-1) indicators for whether person i is to receive item j, and p, a bounded continuous variable used to track the maximum envy between pairs within the group. This mix of binary and continuous variables define this as a mixed integer linear program (MILP). The complexity of MILPs is vastly greater than that of their LP counterparts since MILPs are non-convex. MILPs exist in the NP-Hard class of problems and thus no algorithm can be created to solve them in polynomial time, making runtime a serious factor to consider throughout this work.

The first set of constraints in the formulation (2) enforces that each item is allocated once and only once by ensuring that the sum over all people for each item of the corresponding decision variables is equal to one. This takes the form of equating the summation of $|P|$ number of binary variables for each item $i \in I$ to one, where P and I are the set of all people and set of all items respectively. Due to the equality constraint, this set of constraints is binding and must be satisfied in any optimal allocation determined.

The second set of constraints found in our formulation (3) ensures that the defined objective value, p, is correct. This is done by noticing that the value of any person's bundle from the perspective of someone else can be modeled using the sum of the second person's valuation of the first person's bundle items, leveraging the additivity of values. A person's bundle contents are shown by which indicator variables, $x_{i,j}$, for that person are one, thus when the second person's value for each item is multiplied by the indicator's for the first person's bundle and summed, the second person's value of the first person's bundle is obtained. To obtain the value of someone's own bundle this same process is repeated except the set of indicators is chosen for the person measuring value. By subtracting these two bundle values and setting p to be greater than or equal to the result, only one of these differences – the maximum difference – will bind the objective. Thus, p is obtained as defined.

Constraints (4) and (5) bound p to be between zero and one. This upper-bound is established due to normalization of values, which implies the maximum possible value of any bundle is one and therefore the maximum possible value of p is also one. This value would be obtained in the case where one person receives all the items of the allocation. p is also non-negative to avoid the concept of negative envy. Furthermore, (6) is to ensure that $x_{i,j}$ is binary.

Table 1: Summary of Results of MILP for Approximate Envy-Freeness

Solver Status	Result Counts
Optimal Solution Found	308
Optimal Solution Indicated, but Error Likely	16
Constraints Cannot Be Satisfied	0
Objective Can Be Improved Without Limit	0
Stopped by a Limit (Such As on Iterations)	76
Stopped Due to Solver Error	0
Total Number of Results	400

2.2.3 Model Implementation and Solving Methods

The formulation is implemented as a .mod file in *AMPL*, found as "pEF.mod" in the repository. A .run script is then utilized to evoke the *CPLEX* solver, found as "pEF.run". The solver is run on two small examples for which analytical solutions can be derived, in order to measure the accuracy of the implementation, then, once deemed accurate, the implementation is run for all generated problem cases.

The first example consists of two people and two objects, and the second is composed of four people and five objects. Item valuations are assigned for both examples and then normalized. These values, along with the descriptions of the sets of people and items are stored in .dat files, found in the files "simple.dat" and "larger.dat" respectively. In order to enable somewhat modest total runtimes for the generated examples, a callback is utilized for *CPLEX*, limiting the runtime to a maximum of five minutes.

2.2.4 Results and Analysis

For the simpler example, the best objective value was found to be 0.5, with a corresponding allocation of person one receiving item one and person two receiving item two. These results are reflected in the file "simple.txt". For the larger example, the best objective value found was 0.0465, with a corresponding allocation of person one receiving item one, person two receiving item two, person three receiving items three and five, and person four receiving item four. Similarly, these results are reflected in the file "larger.txt". A summary of solver outcomes for the 400 generated data files can be found in Table 1 and the totality of collected data can be viewed in the virtual notebook instance or in the file "results.txt".

In order to aptly model the multi-dimensional nature of the solver's resultant data, numerous kinds of visualizations are made – each of which help to clarify the underlying patterns governing the change in p-envy-freeness, as well as, the change in solver run-time. Throughout these visualizations, the findings of Dickerson et al. (2014), in addition to Manurangsi and Suksompong are reflected to a certain extent [8] [11, 12].

Bar charts for both the log approximate envy-freeness value and log solver elapsed time for differing numbers of people, grouped by differing numbers of items, are the first visualization considered and are found in appendix A.1.1. In order to provide accurate analysis of the problem, all combinations of number of people and number of items considered must be shown. Thus, the charts are broken into four sections, each having five grouped bar sections,

to enable better readability. The number of bars indicate the number of items for which the variable of interest was non-zero. It can be seen that the number of items needed before reaching an envy-free allocation increases with the number of people, albeit at a faster rate, with twenty items still representing a positive p when the number of people is equal to eleven. This demonstrates a loose lower-bound on the existence of envy-free solutions of $n \geq 2m$ where m is the number of people and n is the number of items. A connection can be made to the concept of whether an allocation is balanced and that allocation's propensity to have low envy. The definition of a balanced allocation is that the different bundles of items allocated all have the same cardinality [3]. Most clearly visible for $n > 10$, p gradually decreases up to the first balanced allocation, then increases modestly directly afterwards, followed by another decrease towards the second situation of a potentially balanced allocation. This demonstrates that p is related to the closeness of m and n such that when the magnitude of their difference is large, p is large, and when n is a multiple of m, p is near or at a local minimum.

A look at the solver elapsed time yields insight to the complexity of related optimization problems for different instances. Until n becomes greater than five, consideration of runtime is neglible since for each instance less than a tenth of a second is needed. Starting at $n = 6$, a degree of variation among the instances begins to increase and when $m \geq 10$, solver callbacks begin to be used (the instance takes at least 300 seconds to solve). For these larger instances of m, runtime generally increases up to a point near a multiple of m, sharply decreases for an instance, then sharply increases again. Overall, solver elapsed time also increases as m and n grow.

A better notion of what the set of all allocation p-envy-free values contains as well as the associated set of run-times is received from consideration of heat maps and triangular surface plots, found in appendix A.1.2 and A.1.3. This is where the findings of Dickerson et al. (2014), in addition to Manurangsi and Suksompong become more clear in this work [8] [11, 12]. A distinct change in the rate at which p changes is made across the boundary where the number of items becomes greater than or equal to the number of people. In addition, the elapsed solver time is highest around the start of this boundary – when the number of people is equal to the number of items – and when the number of people and number of items is largest.

By considering the mean allocation p-envy-free value across all items for the different number of people a somewhat logarithmic relationship emerges. Whereas, when the mean is taken across all people for the different number of items a seemingly reciprocal function in the form of a hyperbola is clear. However, when mean solver elapsed time is considered, both cases form what seems to a periodic function.

Finally, parallel categories diagrams more clearly reveal the inverse relationship between number of people and number of items for p, as well as, the direct relationship between them for elapsed solver time.

2.3 p-Envy-Freeness Up to One Item (pEF1)

2.3.1 Problem Definition Alterations

This problem can be considered as a relaxation of the pEF problem (2.2) as it loosens the notion of envy-freeness by considering associated envy values when up to one item is removed from the bundles. For all intents and purposes, this serves to mean that when a given person is measuring the value of someone else's bundle they do not consider what they think to be their own favorite item within that bundle when valuing it. This simple removal of the highest valued item in the set has quite enormous implications for both the extent of the existence of envy-free allocations as well as computational resources needed to find optimal allocations.

This problem maintains the assumptions made in the original problem concerning item indivisibility and value additivity. p-envy-freeness of up to one item is as follows [5]:

$$p = \max_{i,j,k} \left(\mathrm{v}_i(B_j \setminus k) - \mathrm{v}_i(B_i) \right) \mid i, j \in P, k \in B_j : i \neq j$$

where, as in case of pEF, the overall objective of the problem takes the form of:

$$\min p$$

2.3.2 Mixed Integer Program Formulation and Explanation

$$\begin{aligned}
\text{minimize} \quad & p && && (1)\\
\text{subject to} \quad & \sum_{i \in P} x_{i,j} = 1 && \forall\, j \in I && (2)\\
& y_{i,j} - x_{i,j} \leq 0 && \forall\, i \in P, j \in I && (3)\\
& \sum_{i \in P} \sum_{j \in I} y_{i,j} = \psi \cdot (|I| - |P| + 1) && && (4)\\
& \sum_{j \in I} (v_{i,j} y_{k,j} - v_{i,j} x_{i,j}) \leq z && \forall\, i, k \in P : i \neq k && (5)\\
& z - 1 \leq 0 && && (6)\\
& z \geq 0 && && (7)\\
& x_{i,j}, y_{i,j}, \psi \in \{0, 1\} && && (8)
\end{aligned}$$

In order to model this problem, new decision variables are introduced. These variables take the form of binary (0-1) variables indicting the items left in the bundles after the optimal person has removed one item. Similarly to the last formulation, these variables are a harsh necessity. They add complexity to the problem, while also serving a crucial role of aptly modeling an item's indivisibility.

The constraints of the pEF problem require modification to model this problem. Firstly, a new constraint is added in order to track item set continuity between the original allocated

Table 2: Summary of Results of MILP for AEF1

Solver Status	Result Counts
Optimal Solution Found	400
Optimal Solution Indicated, but Error Likely	0
Constraints Cannot Be Satisfied	0
Objective Can Be Improved Without Limit	0
Stopped by a Limit (Such As on Iterations)	0
Stopped Due to Solver Error	0
Total Number of Results	400

sets, and the new, one item removed sets (3). This constraint takes the form of ensuring that each $y_{i,j}$ is less than or equal to each $x_{i,j}$, forcing items to either persist or drop from the bundles. Put another way, it ensures that no swapping of items occurs and that the set, $\{B_i \mid B_i = \cup_j y_{i,j} \quad \forall j : y_{i,j} = 1\}$ has a cardinality less than or equal to the cardinality of $\{C_i \mid C_i = \cup_j x_{i,j} \quad \forall j : x_{i,j} = 1\}$. Next, another constraint is added to ensure that the total number of considered items is equal to the total number of items minus the number of people plus one (4). This is considered as the cardinality of $\{\cup_i B_i\} = \{\cup_i C_i\} -$Number of People$+1$. The one is added because for an optimal pEF1 allocation there will be a single observer with a maximum envy value for the group and from this observer's perspective every other person besides themselves will have one item removed from their bundles. The constraint for allocating all items persists from the last formulation (2), and the constraint to find the maximum pair-wise envy of the group is modified slightly (5). Instead of using the $x_{i,j}$ indicator in consideration of the other person's set value, that indicator is now replaced by $y_{i,j}$. Lastly, an additional binary parameter is created to signify whether there are more items than people for a particular generated problem (ψ). This proves necessary because the cardinality of B_i dictated by the model constraint can at least be 0, and thus if there exists more people than items, it takes the cardinality of B_i to zero.

2.3.3 Program Generation and Solving Methodologies

For this problem the same generated datasets are reused with a changed *AMPL* model. This model is once again run through the *CPLEX* solver and results generated for the two examples as well as the 400 generated instances. A summary of the solver statuses for the different instances can be found in Table 2, with full results found in "results.txt" in the corresponding directory.

2.3.4 Results and Analysis

For the simpler example of the two, the best objective value was found to be 0, with a corresponding allocation of person one receiving item two and person two receiving item one as shown in "simple.txt". For the larger example, the best objective value found was 0, with a corresponding allocation of person one receiving item five, person two receiving items two and three, person three receiving item one, and person four receiving item four, shown in the file "larger.txt".

The notion that pEF1 is a rather liberal relation of the pEF is supported throughout the created visualizations. There seems to be clear jump in rutime when the number of items, n, is one less than the number of people m. This increased runtime is then maintained throughout the rest the instances for a certain m. Furthermore, the opposite side of the boundary where $m = n$ has increased runtimes as opposed to pEF. The peak rutime is seen when m and n reach their respective peaks, and the boundary of transitioning rate change in p is maintained but reflected. When looking at the mean runtime across all items, a negative quadratic or possible periodic relationship is seen. Whereas when looking at the mean rutime across all point an exponential relationship is found. These visualizations can all be found within appendix A.1.1.

2.4 p-Envy-Freeness with a Subsidy (pEFS)

2.4.1 Problem Definition Alterations

The last problem we consider is p-envy-freeness with an added subsidy in the form of a divisible cash sum. This is a particularly interesting problem since it is fair division of divisible and invisible items. Leveraging work found in [4] the following change is made to the definition of p for a subsidy S:

$$p = \max_{i,j,k} \left[\frac{\sum_{i,k \in P} [(v_{ij} x_{kj} + c_k S) - (v_{ij} x_{ij} + c_i S)]}{\sum_{j \in P} (v_{ij} + S)} \middle| i, j, k \in P : i \neq k \right]$$

where, as before, the overall objective of the problem takes the form of:

$$\min p$$

It is important to consider the added cash as a subsidy since the situational mechanics take the form of paying off binding envy values. That is to say, in situations where the optimal pEF allocation produces positive p values for people, these people can then be paid a proportion of a cash sum in order to compensate for their perceived lower valued bundle. Thus, there exists a lower bound on the total subsidy value such that an allocation produces a completely envy free situation. Therefore, as this problem is formulated and solved, it is interesting to consider how that lower total subsidy value can be determined and further, what insight the determination of this lower bound can provide for pEF allocations.

2.4.2 Mixed Binary Integer Program Formulation and Explanation

$$\begin{aligned}
\text{minimize} \quad & p && && (1)\\
\text{subject to} \quad & \sum_{i \in P} x_{ij} = 1 && \forall j \in I && (2)\\
& \sum_{i \in P} c_i = 1 && && (3)\\
& \frac{\sum_{i,k \in P}(v_{ij}x_{kj} + c_k S) - (v_{ij}x_{ij} + c_i S)}{\sum_{j \in P} v_{ij} + S} - p \leq 0 && \forall i,j,k \in P : i \neq k && (4)\\
& p - 1 \leq 0 && && (5)\\
& c_i - 1 \leq 0 && \forall i \in P && (6)\\
& p, c_i \geq 0 && \forall i \in P && (7)\\
& x_{i,j} \in \{0,1\} && && (8)
\end{aligned}$$

The formulation of the pEFS problem inherently must contain a set of continuous variables so that the divisibility of the subsidy can be modeled correctly. Furthermore, the added subsidy increases the total possible value of any one person's set, thus the subsidy must be added into the formulation before normalization of the data takes place.

The constraint from pEF involving allocation of all items begins the formulation (2). Next, there is a constraint to ensure that the entirety of the subsidy is allocated (3). This takes the form of a summation over all people for the continuous variable of the proportion of the subsidy each person is to receive in the allocation. This sum must be equal to one since they are proportions. Finally, the constraint to find the pairwise maximum p of the group must be modified to agree with the above definition. This constraint mirrors that of pEF, except that the value of a certain person's bundle is added to the proportion of cash that person receives. Furthermore, normalization of the valuations is built into this constraint by dividing the total – indivisible items and subsidy proportion – values of the bundles by the total possible value of the allocation.

2.4.3 Program Generation and Solving Methodologies

For this problem, due to the nature of the high run-time of CPLEX for these problems, only combinations of one through 15 numbers of people and numbers of items are considered. However, for each of these combinations cash values from \$50 to \$1200 with \$50 increments. The new formulation is modeled in *AMPL* and solved with *CPLEX* using the corresponding subset of the original generated data and cash amounts.

2.4.4 Results and Analysis

In the problem the examples are utilized differently than in previous sections. For the smaller example, the same data is run, but with the inclusion of a cash subsidy of \$3000. The results found for this example are that person one gets item two and the entire proportion of cash, whereas person two gets item one. Next, the model formulation is applied to the larger

Table 3: Summary of Results of MILP for AEFS

Solver Status	Result Counts
Optimal Solution Found	4186
Optimal Solution Indicated, but Error Likely	883
Constraints Cannot Be Satisfied	0
Objective Can Be Improved Without Limit	0
Stopped by a Limit (Such As on Iterations)	331
Stopped Due to Solver Error	0
Total Number of Results	5400

example, but in this case the goal is to see the minimum amount of cash required to induce a completely envy-free allocation. To determine this, the value of cash is incremented by 1¢ until p was equal to 0. The resultant value of cash needed was \$199.78.

Next, the pEFS model formulation was applied to the randomly generated data. This data proved to be a visualization challenge as important result variables consist of the number of people, the number of items, the p-envy-free value, the solver elapsed time, and the subsidy amount. This combination of variables manifests as generally four-dimensional data to visualize as p can be visualized apart from run-time. The plots and diagrams utilized for this purpose specifically come in the form of an interactive triangular surface plot of which can be dynamically changed to reflect different subsidy levels, a three-category parallel categories diagram, and a four-dimensional scatter plot representation where different subsidy levels come in the form of different plot markers for 25 randomly selected combinations of number of people and number of items. After consideration of the output of these tools, a few patterns come to light.

The surface plots demonstrate the same overall pattern found in pEF (2.2). Looking at how the surfaces change for different subsidy values it can be seen that the subsidy value acts as a reward to the overall envy-freeness. The subsidy serves to penalize the function mapping number of items and number of people to envy-freeness and scales p down an amount commensurate with its ability to subsidize envy. Elapsed solver time in this case is is a bit harder to parse, however, there seems to be a pattern where the number of cases that take on higher values of runtime increases with the subsidy amount as well as the highest runtimes occuring when the subsidy has to be divided more ways. This makes sense as it requires more time to figure out optimal allocations when there are more people since satisfying everyone requires a search through a larger space.

The parallel categories diagram shows that high runtimes are associated with higher numbers of people and items, regardless of the subsidy amount. Furthermore, the three-dimensional randomly sampled scatterplot shows that runtime is not particularly affected by subsidy amount, and that the affect of subsidy amount is not linearly related to the change in p.

2.5 Strategies to Improve Runtime

Numerous strategies for possibly improving the *CPLEX* solver runtime are introduced here. These strategies are implemented and tested on the pEF problem due to the compelling nature of the earlier results. These strategies include: initializing the solver to a decent (pEF1 problem satisfactory) feasible solution before runtime, setting an upper bound on the objective function, a combination of both setting an upper bound and initializing a feasible solution, and finally, tuning the CPLEX parameters. These strategies were chosen after review of existing recommendations for runtime improvement [7] [10]. The intricacies of the different strategies as well as reasoning and discussion of gathered results is given for each strategy. A note must added, which is that data comparison of any sort in this work has a certain degree of difficulty since the solver was not run on a dedicated machine. Thus, fluctuations in the use of system resources by other processes can add variance to the results. In general, the only strategy presented here that had tangible runtime improvements was tuning the solver parameters. However, it is with hope that these strategies can be further explored and an understanding as to why or why not they decrease runtime can be found.

2.5.1 Upper Bound

In Lipton et al., an upper bound on envy is found to the maximum marginal value that can be contributed by any item [9]. For our case, this amounts to the maximum overall item value of all items. Thus, this can leveraged by finding the maximum value in the pre-processing step and adding this value as the upper bound to the objective in the formulation. This could be easily accomplished by modifying the .mod file in *AMPL*. The reasoning behind adding an upper bound is that this can serve to decrease the size of the associated problem's polytope and corresponding convex hull thus decreasing the search space for *CPLEX* to branch within with hopes of reducing runtime.

2.5.2 Initial Solution

IBM does not specify a specific methodology for generation of a starting solution, but through the application of the common draft based pick system used by most professional sports leagues an initial starting solution that is satisfactory to produce completely envy-free up to one item allocations. This algorithm, of which this paper will refer to as *Round Robin*, is in fact proven to be satisfactory for the pEF1 problem as shown by Caragiannis et al. [5].

The algorithm begins by creating an arbitrary ordering of the people, which in this case we use the natural ordering provided by the person numbering. Next, the first person is allocated their favorite item. Then, the next person is allocated their favorite item amongst the remaining items. This methodology is continued until all items have been allocated. In the situations where there are more people than items, the original ordering of people is simply repeated. For ease of information input and output, this algorithm is implemented in *AMPL* in the form of a command file, of which can be found in the file "round_robin.cmd".

The reasoning behind initializing the solver with a starting solution is that it allows the branching process to being nearer the optimal solution and larger cuts can be generated in the beginning. The Round Robin algorithm is chosen for creating this initial solution for a few reasons. First of all is it intuitive and simple to implement. Secondly, it enables

consistency across different instances. Finally, it satisfies envy-freeness for the case of pEF1. This consideration seemed beneficial since that may be a better solution than an arbitrarily chosen solution. The results of this strategy can be found in the file "start_soln_results.txt".

2.5.3 Upper Bound and Initial Solution

This strategy simply combines both of the previous strategies. The rationale is that if *CPLEX* is able to not only have a smaller search space, but also begin at a point closer to optimality, then larger improvements on runtime can be found. These results can be found in the file "upper_bound_start_soln_results.txt".

2.5.4 Tuned Solver Parameters

The last strategy implemented is tuning the different available options in *CPLEX*. By default, *CPLEX* attempts to use what it thinks to be the best options for the given model, but these results may not always be well tailored. *CPLEX* also includes a built-in option tuner, which was repeated five times for the first 175 datasets. Although there was variability in the options the tuner recommended for different problem sizes, the options that proved consistent were "varsel=4", "heurfreq=-1", and "cutpass=-1". These correspond to branching based on pseudo reduced costs, never applying node heuristics, and no passes permitted when generated cutting planes [6]. The full results of tuning trials run can be found in the "tuning_results" folder of the repository.

To the authors' knowledge tuning these solver parameters is much akin to tuning hyperparamters of machine learning models. Thus, in order to fully ensure optimal parameter choice, a full grid search of all possible parameter combinations must take place. However, due to the time constraints of this research as well as solver run-time, only a brief search through the solver parameter space could be conducted. This search was done by iterating alphabetically through the relevant solver parameters and adding the best option as determined by total elapsed solver time for the first 175 problem instances.

The outcome of this search produced the options: repeatpresolve=2, populate=1, poolreplace=1, pooldual=1, mipbasis=0, varsel=3, mipcuts=1, boundstr=0, coeffreduce=3. These options correspond to running the presolve feature again considering cuts, running the "populate" algorithm after finding a MIP solution, keeping the best solution in cases where the solution pool is more than pool capacity, returning dual values in the solution pool, not returning solver-status for variables to *AMPL*, using strong branching for selecting the next branching variable during branch and bound, moderately generating all cut types, not using bound strengthening, and aggressively reducing coefficients with tiling during preprocessing respectively. Although these options certainly are not the best they could be, they did improve the time to solve the first 175 instances by about 20%.

2.6 Discussion

2.6.1 Comparison Among Envy Definitions

Our results demonstrate that pEF (2.2) has the longest runtime among the different fairness definitions presented. This can be contributed to the fact that pEF1 (2.3) and pEFS (2.4) are viewed as relaxations of the AEF problem.

For pEF1, existence is always guaranteed and multiple allocations exist to ensure envy-freeness [5]. For this reason, pareto optimality (defined by Caraginnis et al. as the existence of no alternative allocation that can make some players strictly better off without making any player strictly worse off) is usually the metric of interest for problems such as this as it enables a singular allocation to represent each instance. However, for the case of pEF1, one thought as to the decrease in runtime can be the fact that the proportion of solutions satisfatory for envy-freeness increases. This increase in turn increases the chance that *CPLEX* will find one of these points, and when one is found *CPLEX* can stop its search.

For pEFS, existence is guaranteed after a certain subsidy amount [4]. When this amount is reached it has the same effect as explained in the last paragraph. However, up to this point, increases in runtime can be explained as an increase to the size of the search space due to the necessity of division of the subsidy.

From a practical point of view, EF1 does not seem like the most useful definition to apply to real world scenarios. EFS seems like the best approach to real world scenarios since people's envy may be removed by using cash. However, if the subsidy amount is not enough to fully pay off all envy, then the resulting allocation will still not be envy free. Envy-freeness overall is quite difficult to achieve, and even harder to compute in many cases. Hopefully advances in research will enable betterment in this area since the practical applications of removing envy when dealing with invidivisible items is quite tangible.

2.6.2 Limitations of Findings

As mentioned earlier in this work, due to the fact that *CPLEX* was not run on a dedicated machine the runtimes found are be somewhat unrealible due to variance of system resources taken up in other processes. Furthermore, the assumptions of value additivity severely limit the practical application of these findings. In addition, the fact that values are independently and identically uniformly distributed also severely limits practicality.

2.6.3 Considerations of Future Research

Further research into the implications of problem phase change could prove to be quite helpful. It is known that as a problem changes from satisfiable to unsatisfiable a drastic increase in problem complexity is found. This is somewhat explained in this work by the mirroredness of runtime about the boundary where the number of people is equal to the number of items for pEF. Looking at the approximate envy, as this work does, as opposed to the existence of envy-freeness, in relation to problem complexity seems like an interesting avenue to better understand the underlying mechanics of fairness under envy.

Appendix A Data Visualizations

A.1 p-Envy-Freeness

A.1.1 Grouped Bar charts

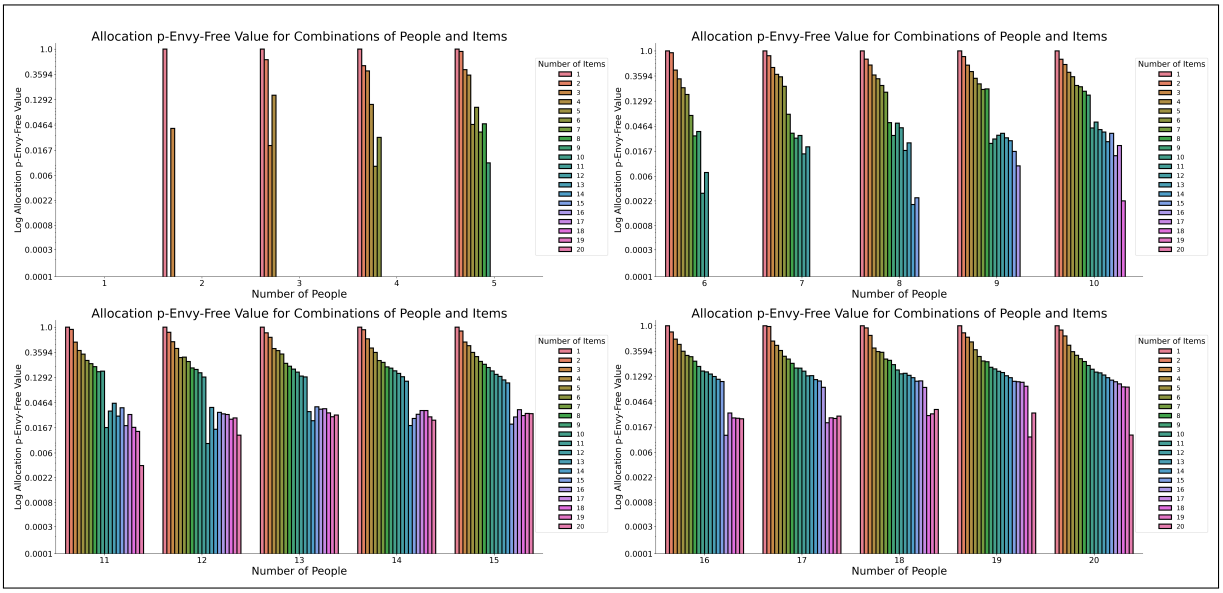


Figure 1: Approximate p Values

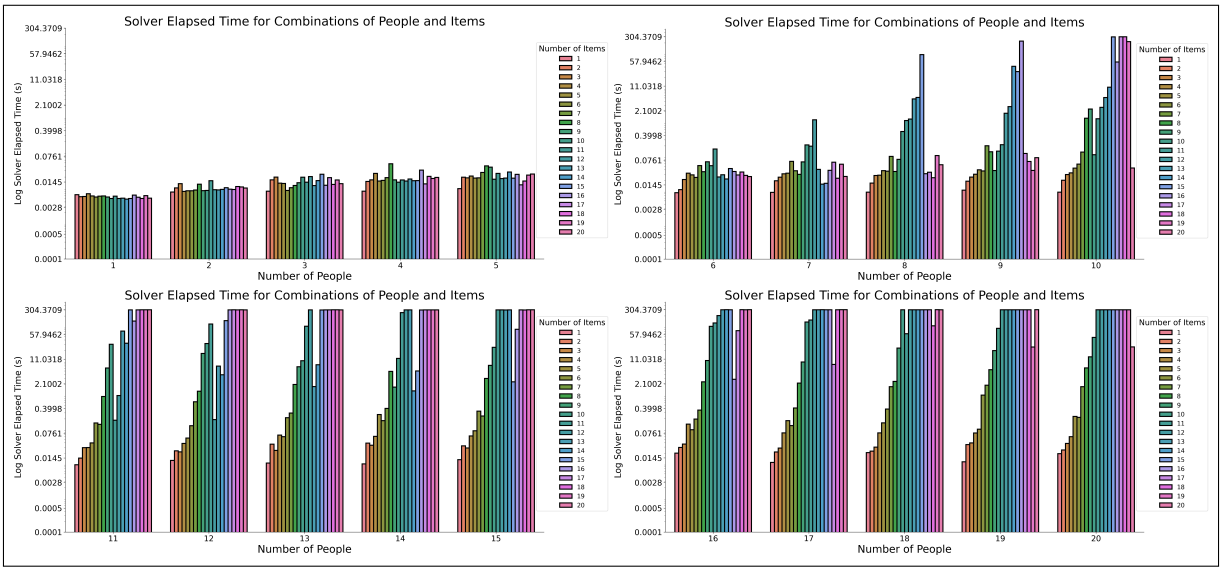
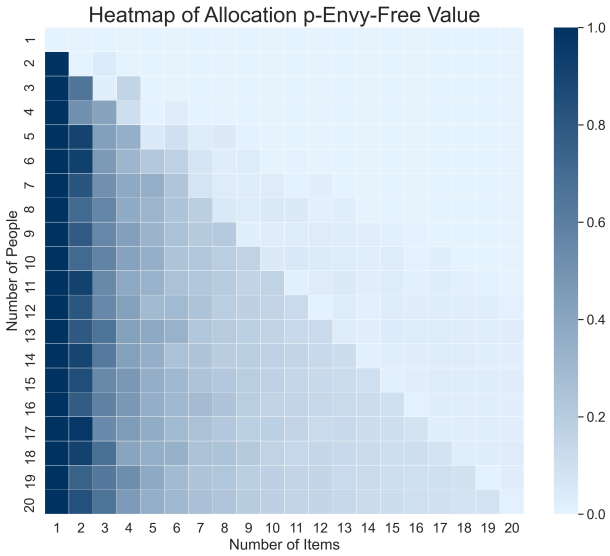
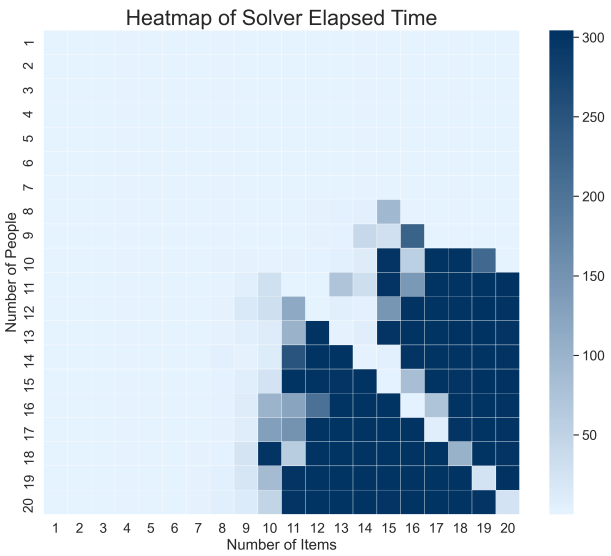


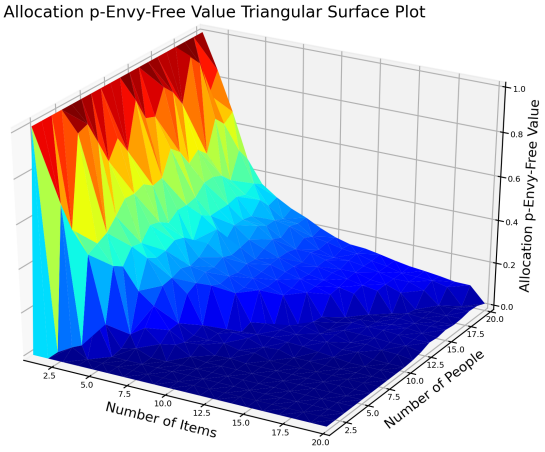
Figure 2: Solver Elapsed Time

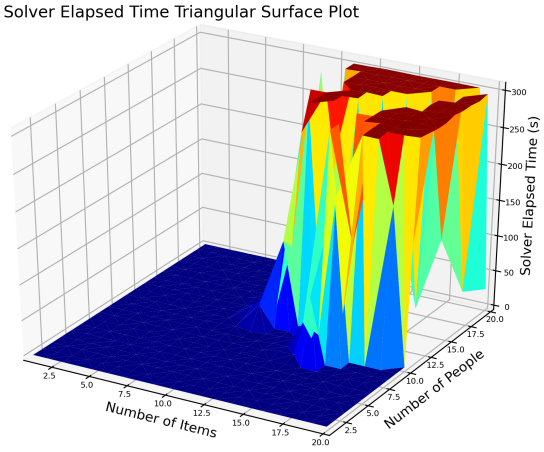
A.1.2 Heat Maps



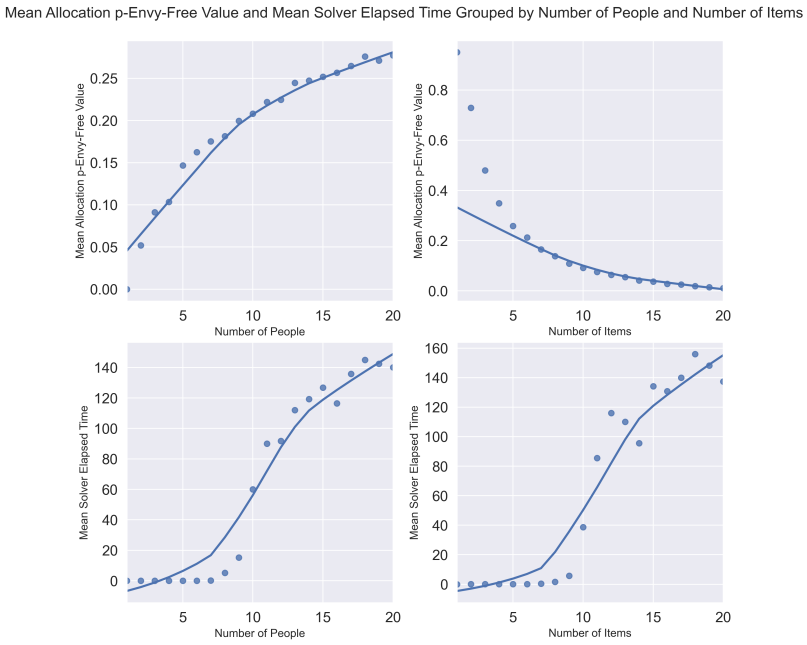


A.1.3 Triangular Surface Plots

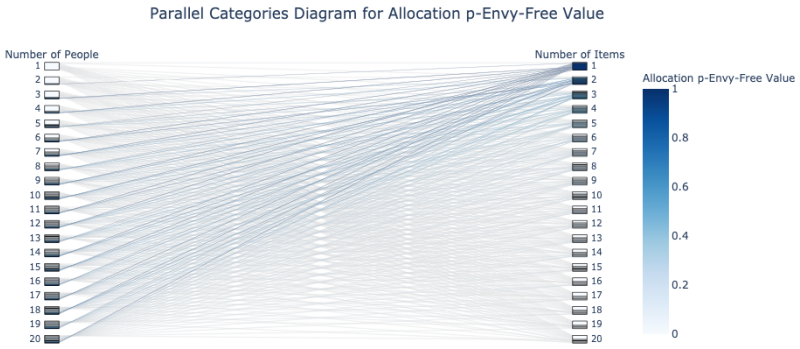


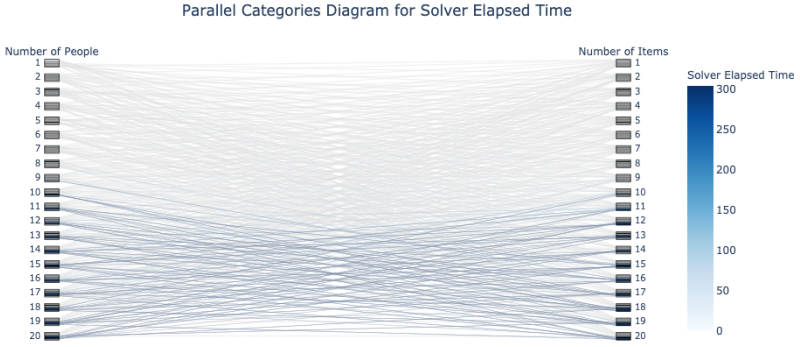


A.1.4 Scatter Plots of Mean Aggregated Data



A.1.5 Parallel Categories Diagrams





A.2 p-Envy-Freeness Up to One Item

A.2.1 Grouped Bar Charts

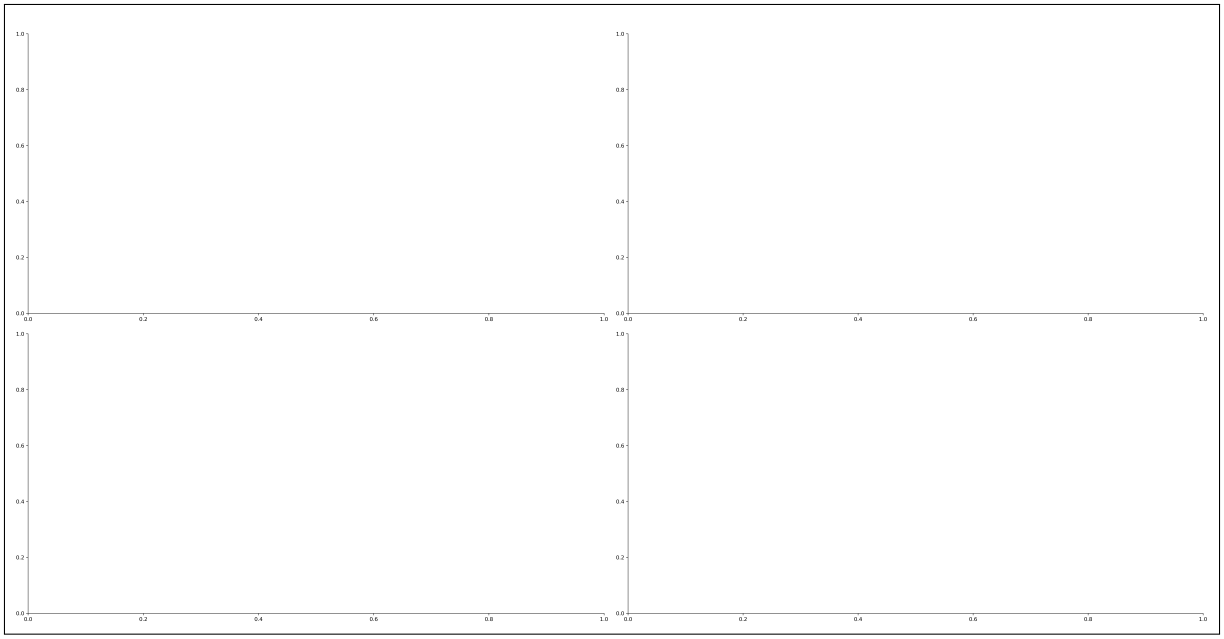


Figure 6: Approximate p Values

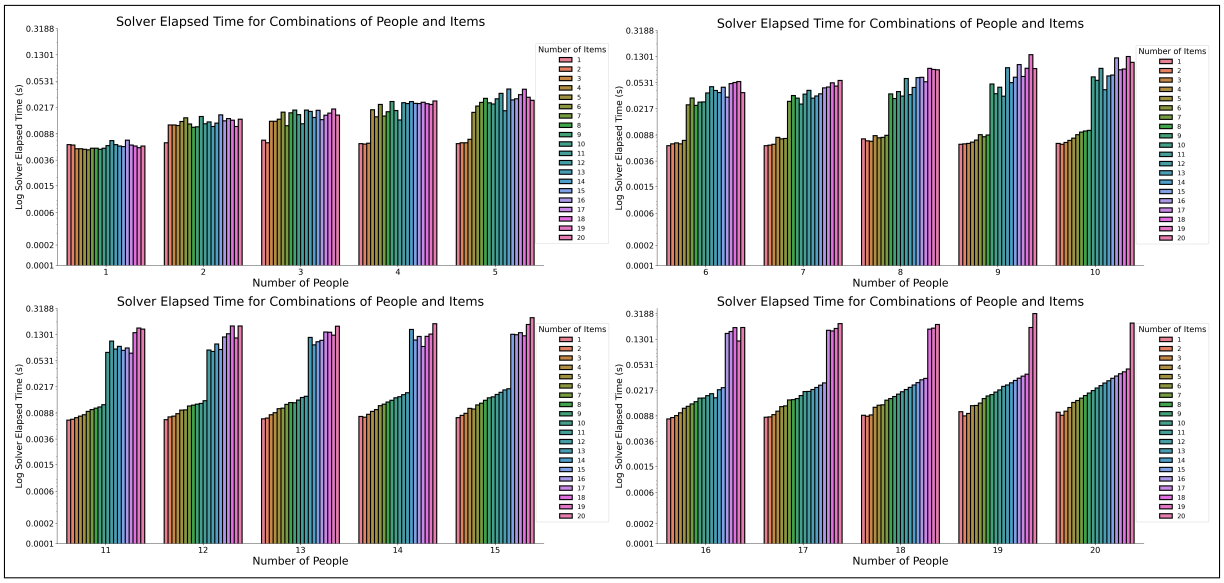
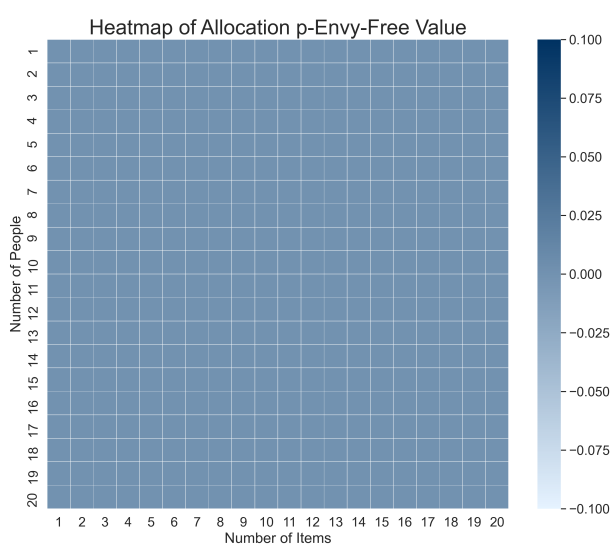
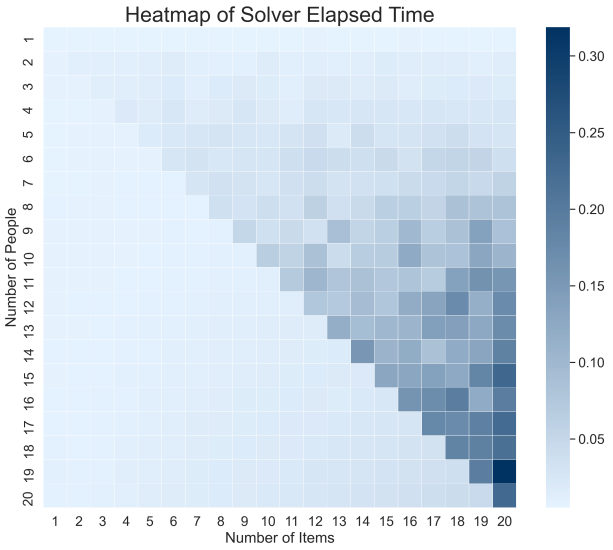


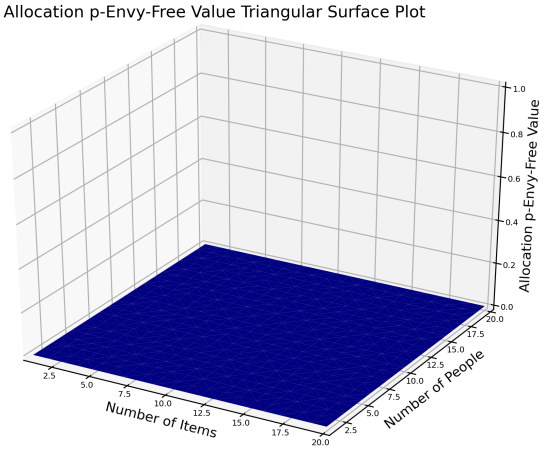
Figure 7: Solver Elapsed Time

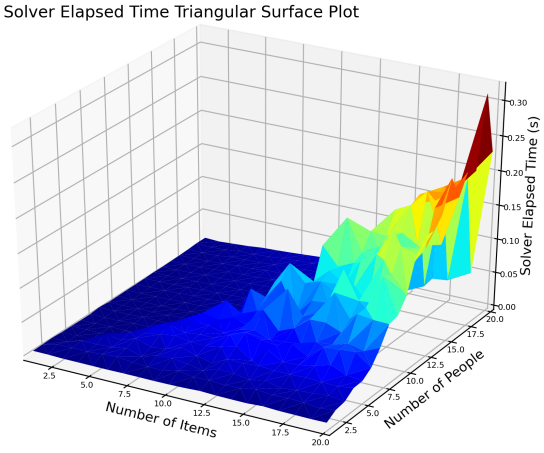
A.2.2 Heat Maps



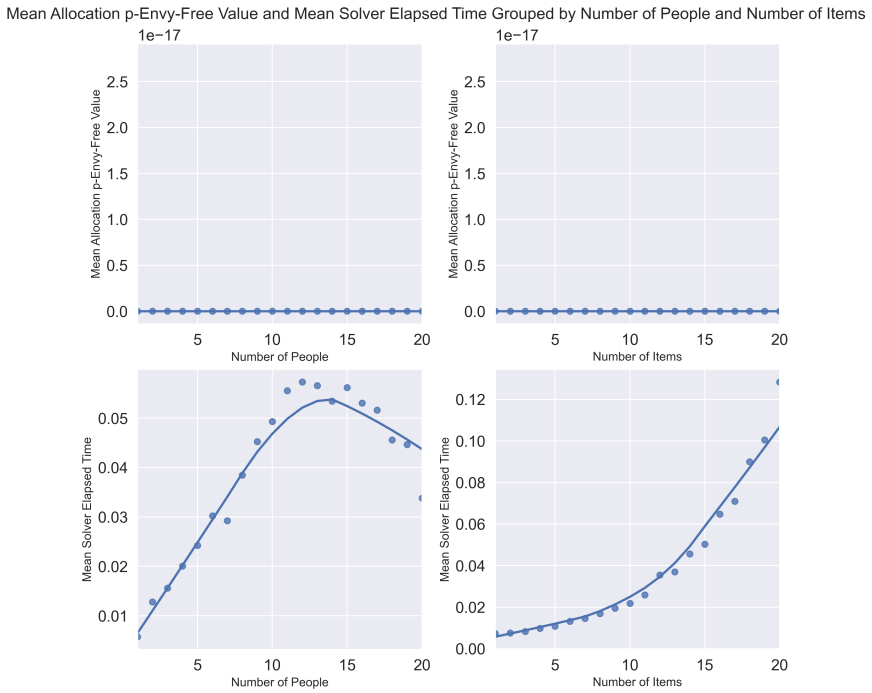


A.2.3 Triangular Surface Plots

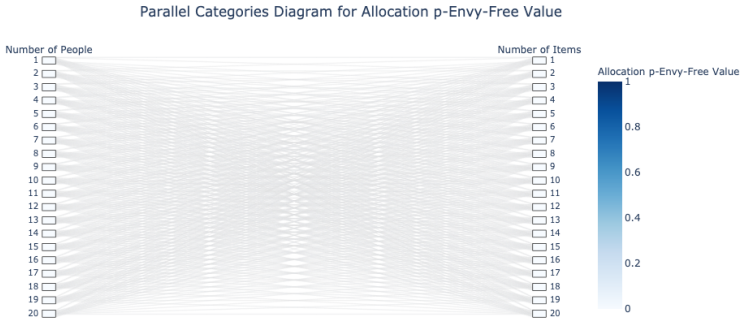


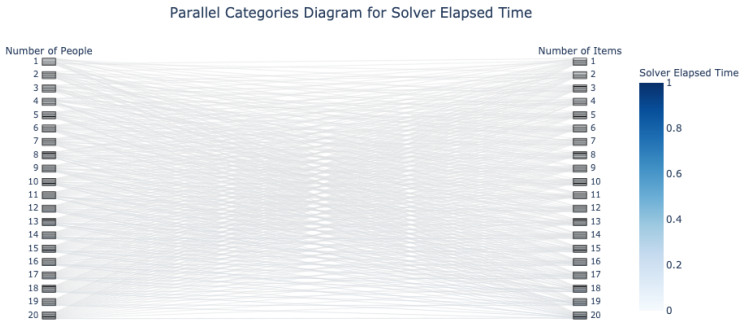


A.2.4 Scatter Plots of Mean Aggregated Data



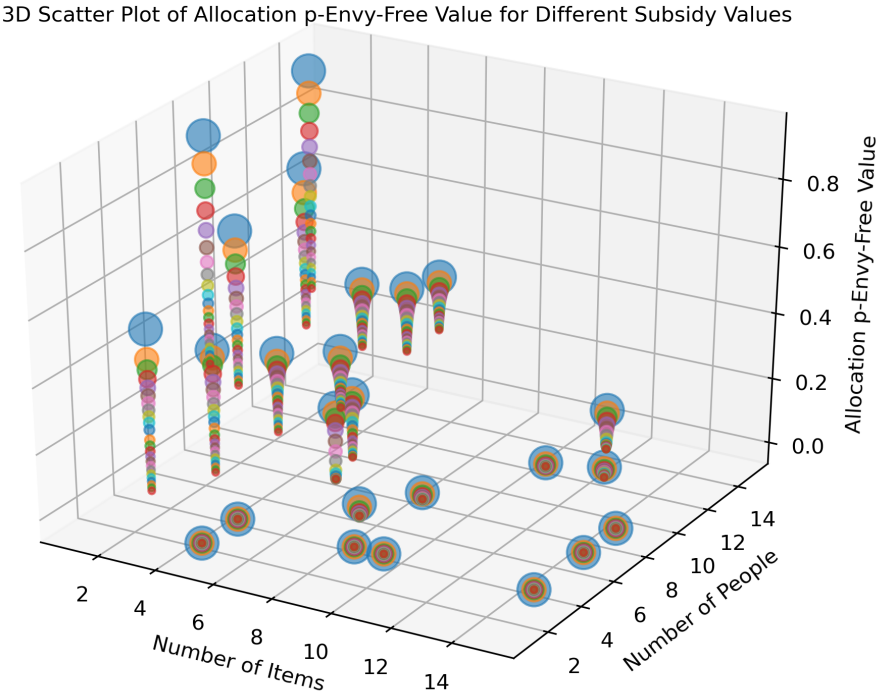
A.2.5 Parallel Categories Diagrams

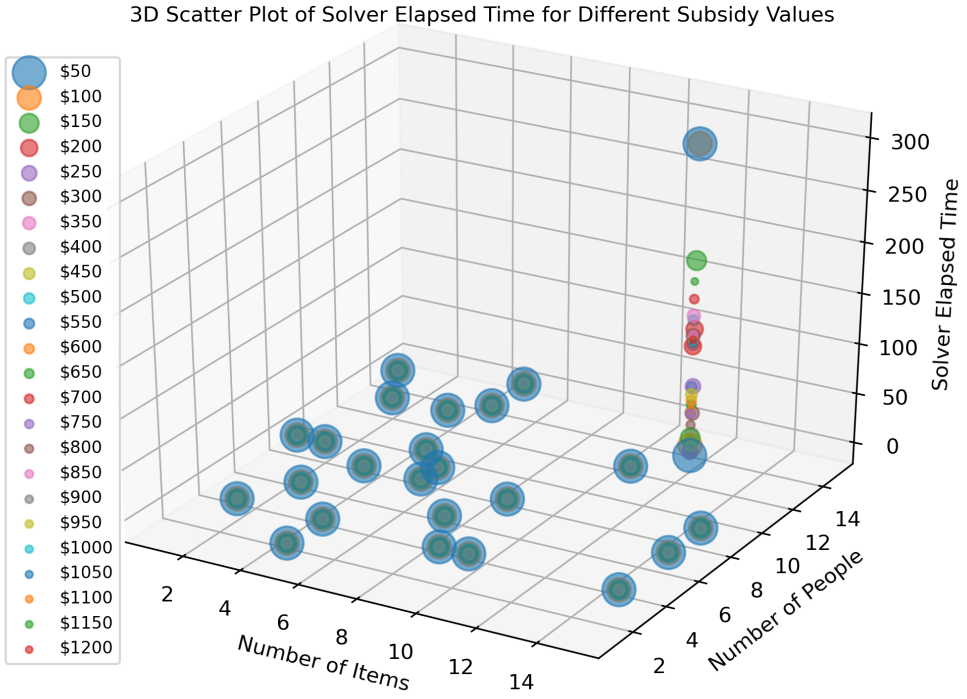




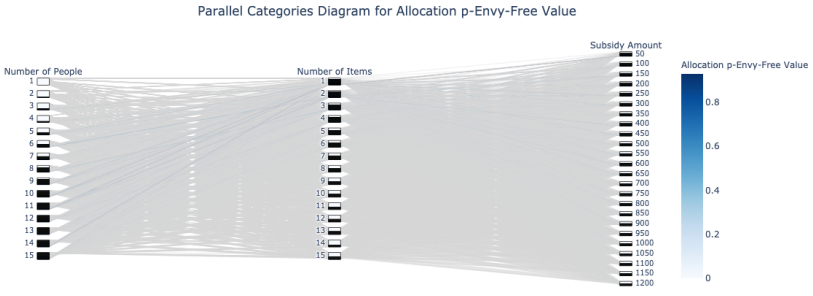
A.3 p-Envy-Freeness With a Subsidy

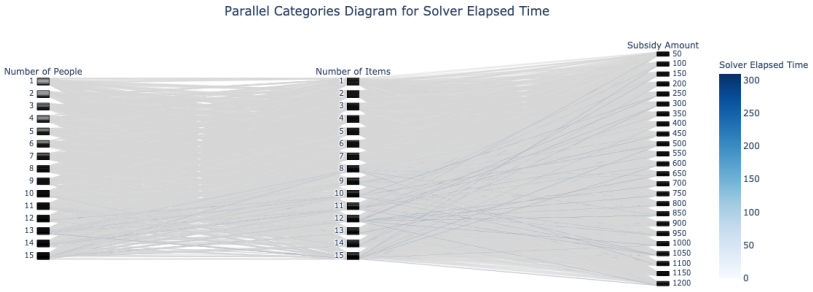
A.3.1 4D-Scatters





A.3.2 Parallel Categories Diagrams





A.4 Strategies to Improve Run-Time

A.4.1 Upper Bound

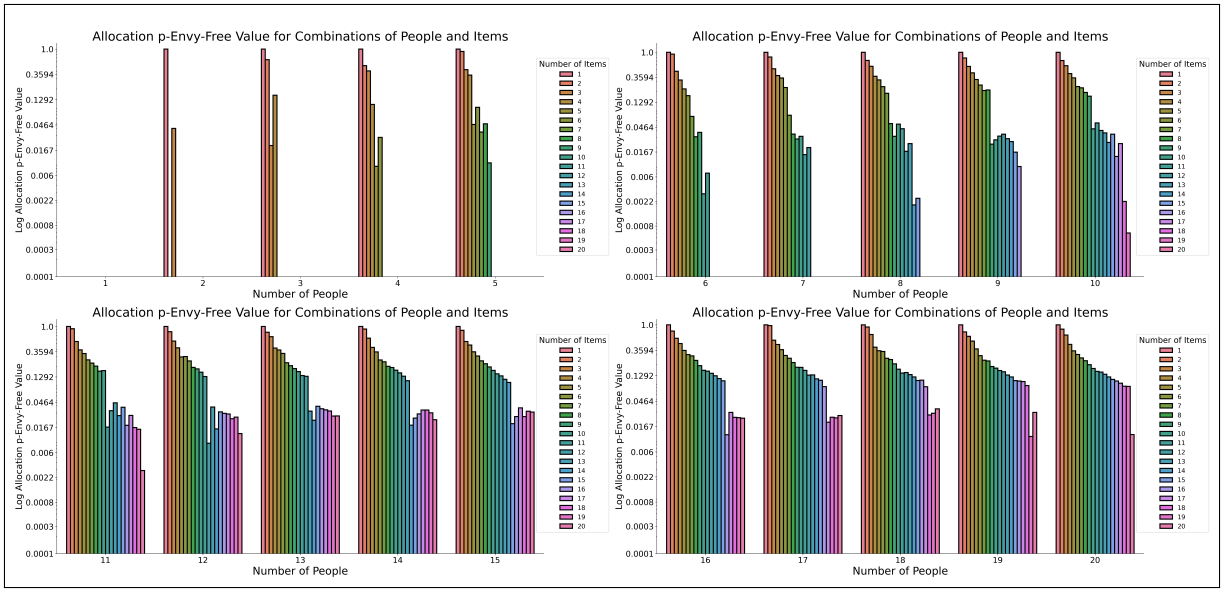


Figure 13: Approximate p Values

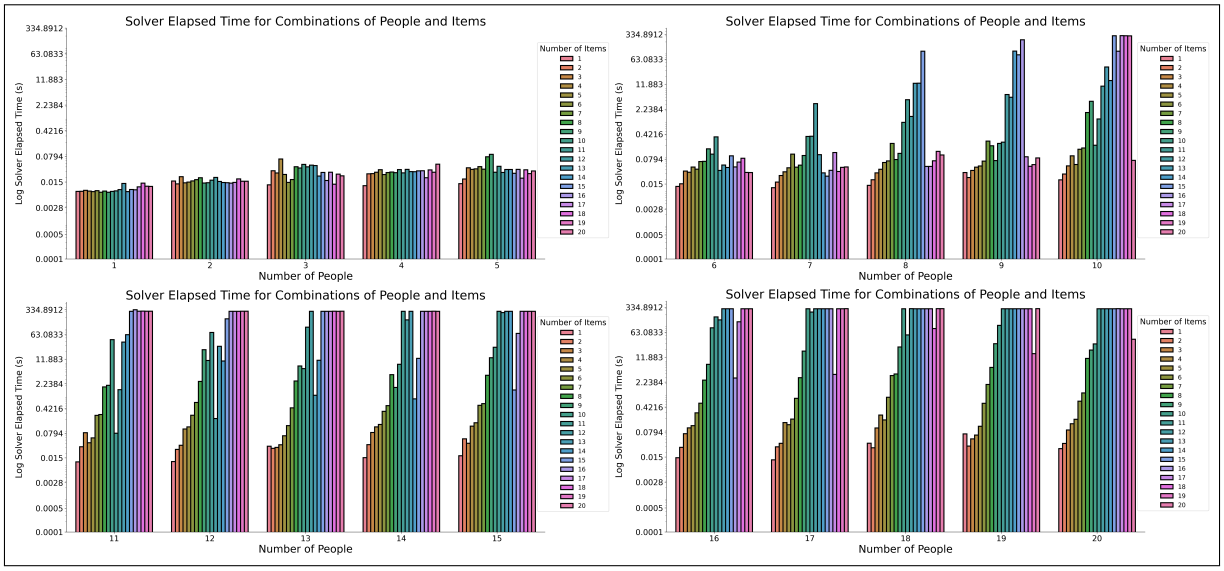
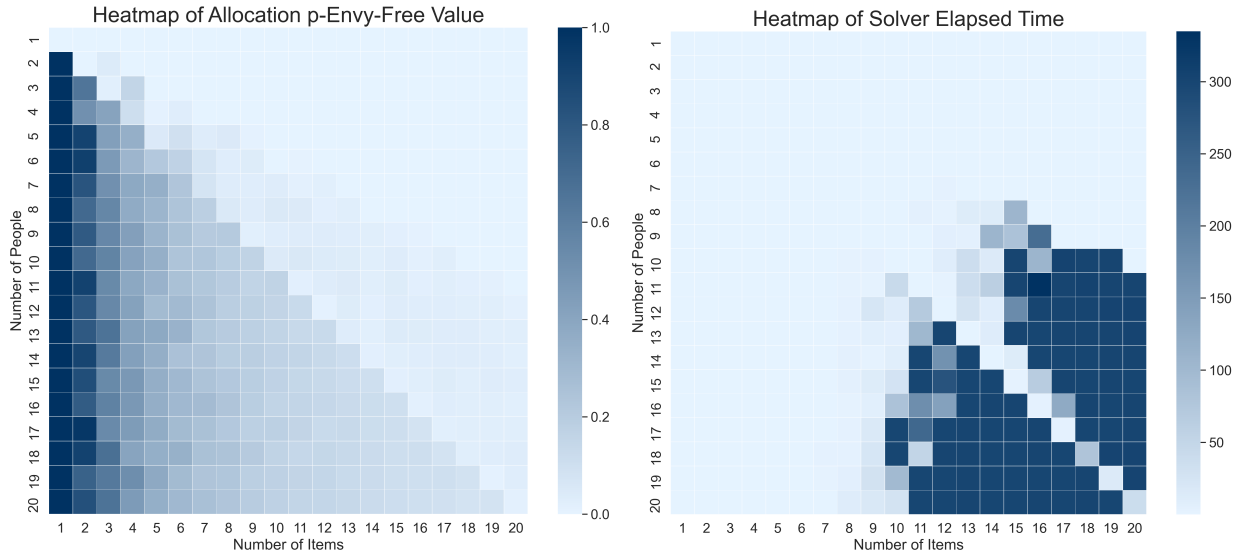
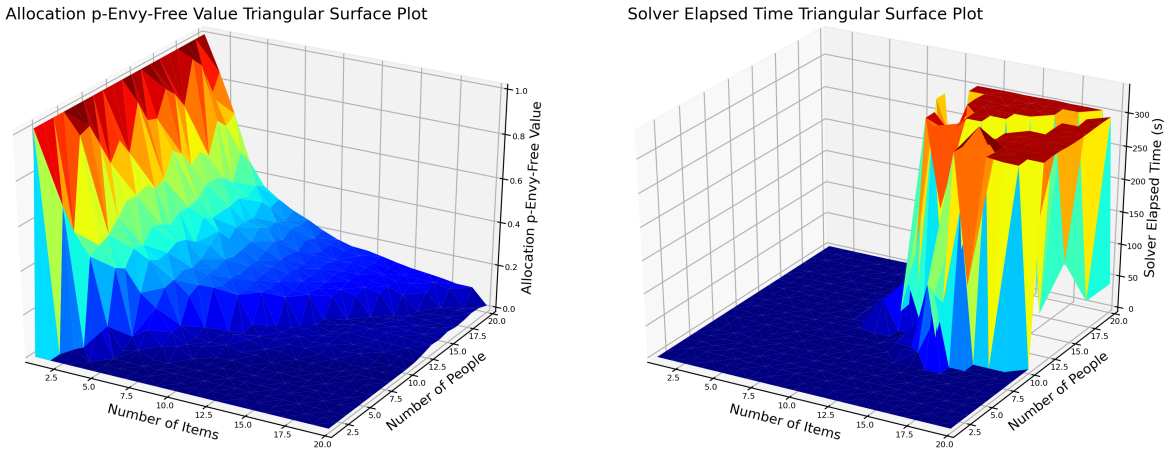


Figure 14: Solver Elapsed Time

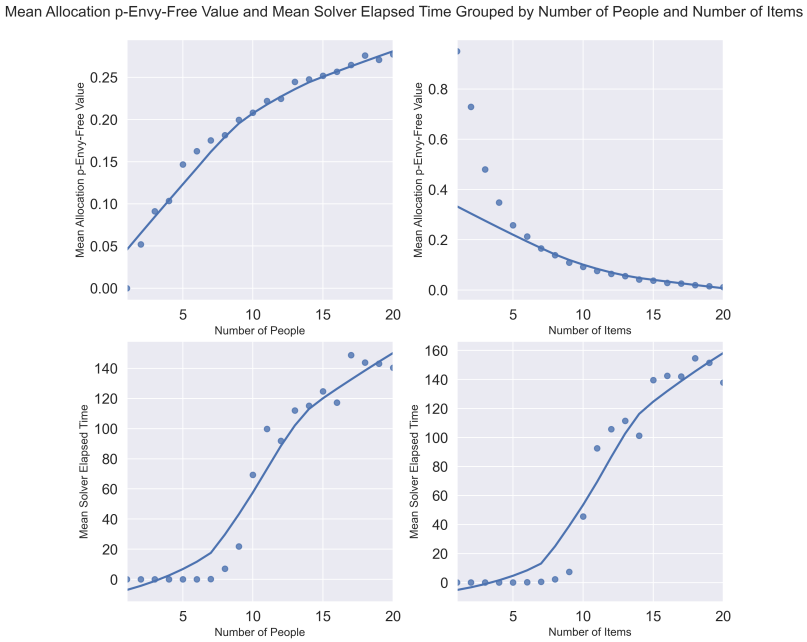
A.4.1.2 Heat Maps



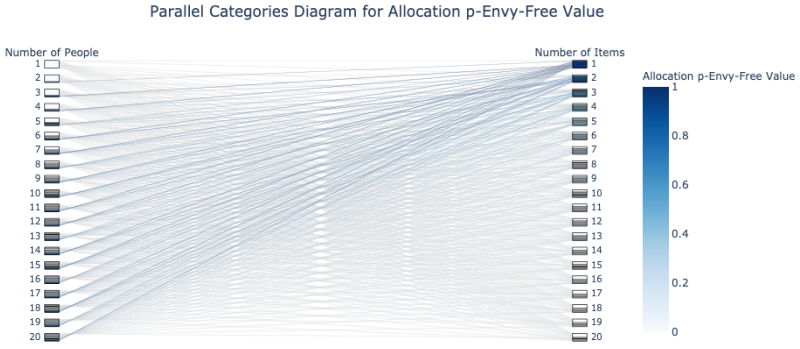
A.4.1.3 Triangular Surfaces

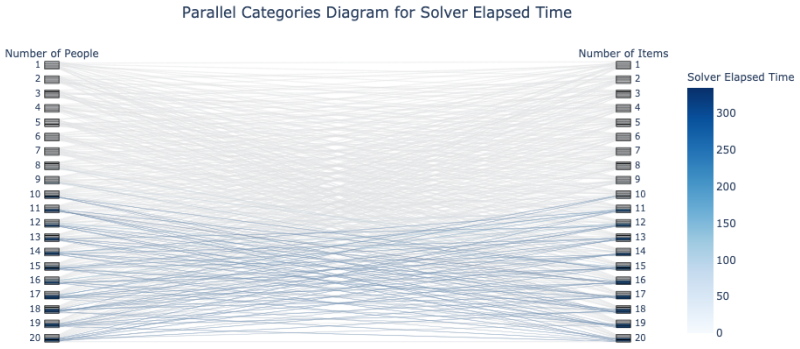


A.4.1.4 Scatter Plots of Mean Aggregated Data



A.4.1.5 Parallel Categories Diagrams





A.4.2 Start Solution

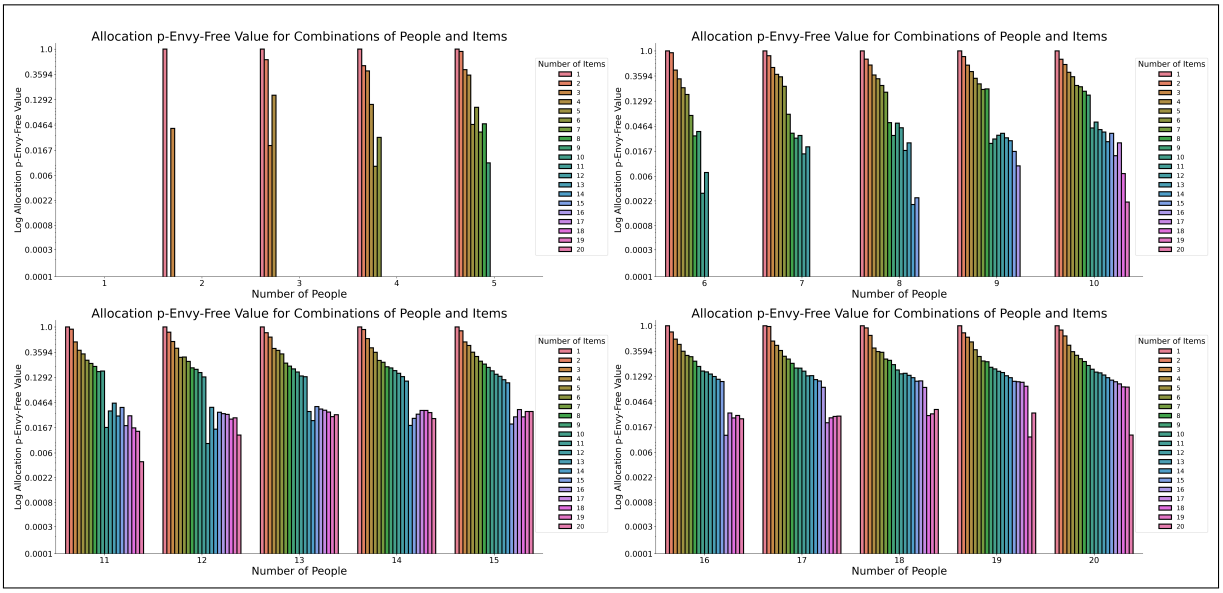


Figure 18: Approximate p Values

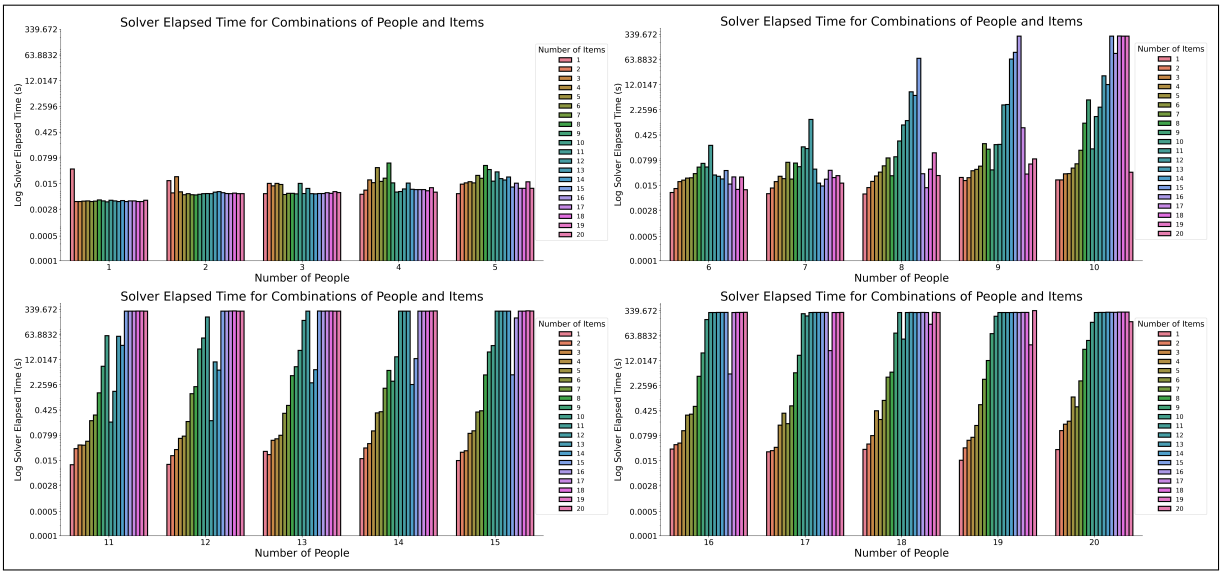
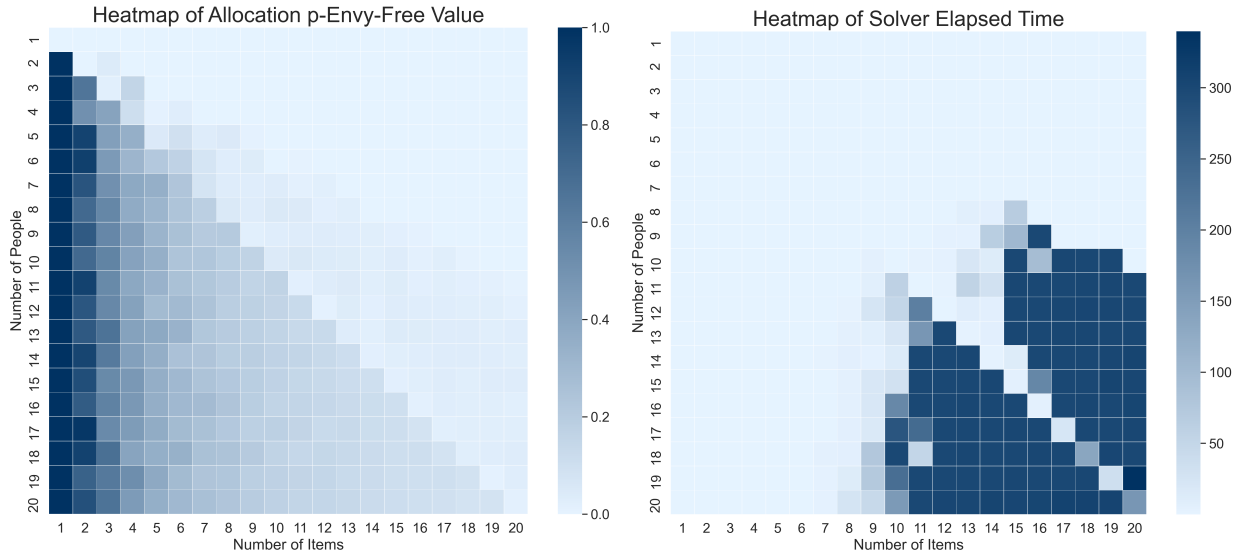
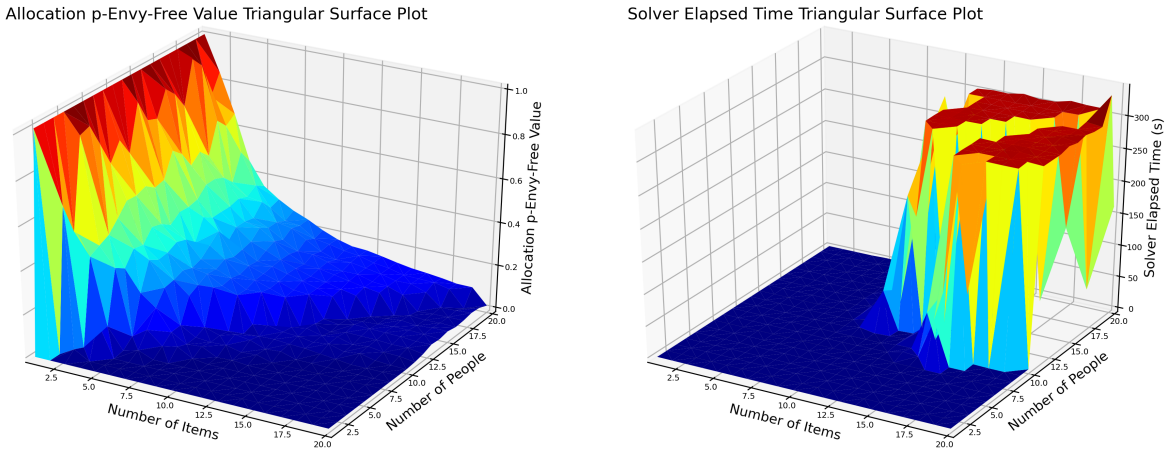


Figure 19: Solver Elapsed Time

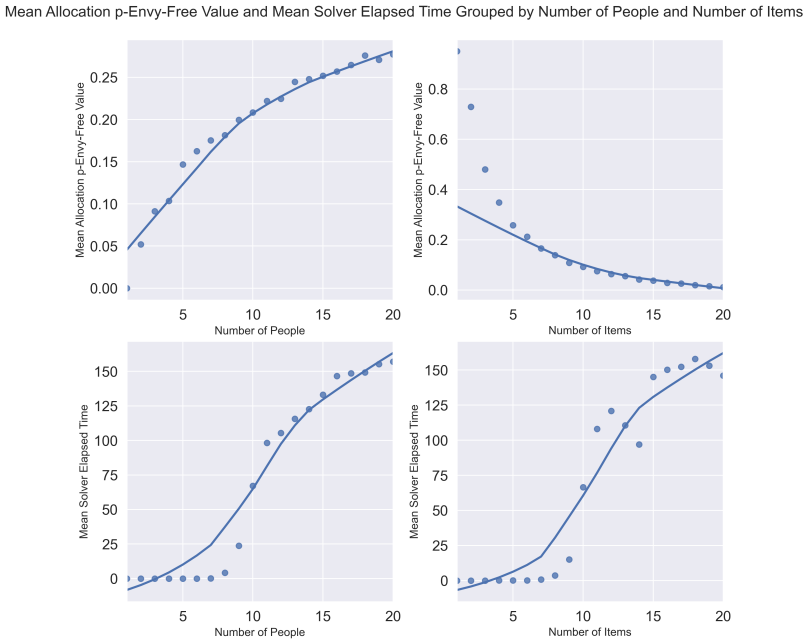
A.4.2.2 Heat Maps



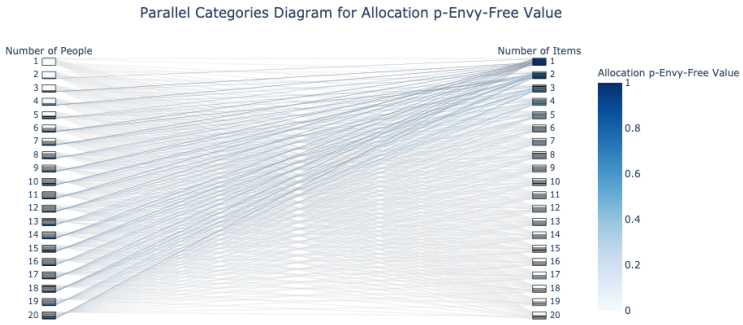
A.4.2.3 Triangular Surfaces

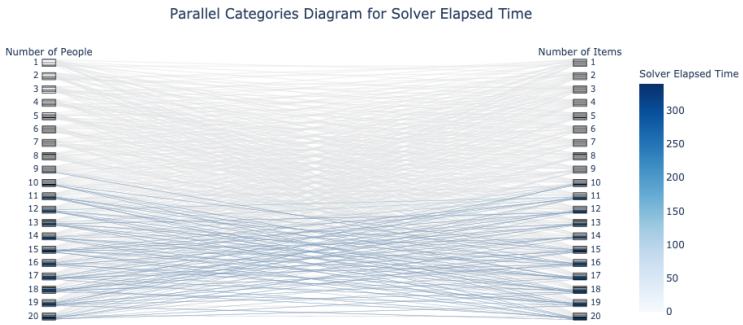


A.4.2.4 Scatter Plots of Mean Aggregated Data



A.4.2.5 Parallel Categories Diagrams





A.4.3 Upper Bound and Start Solution

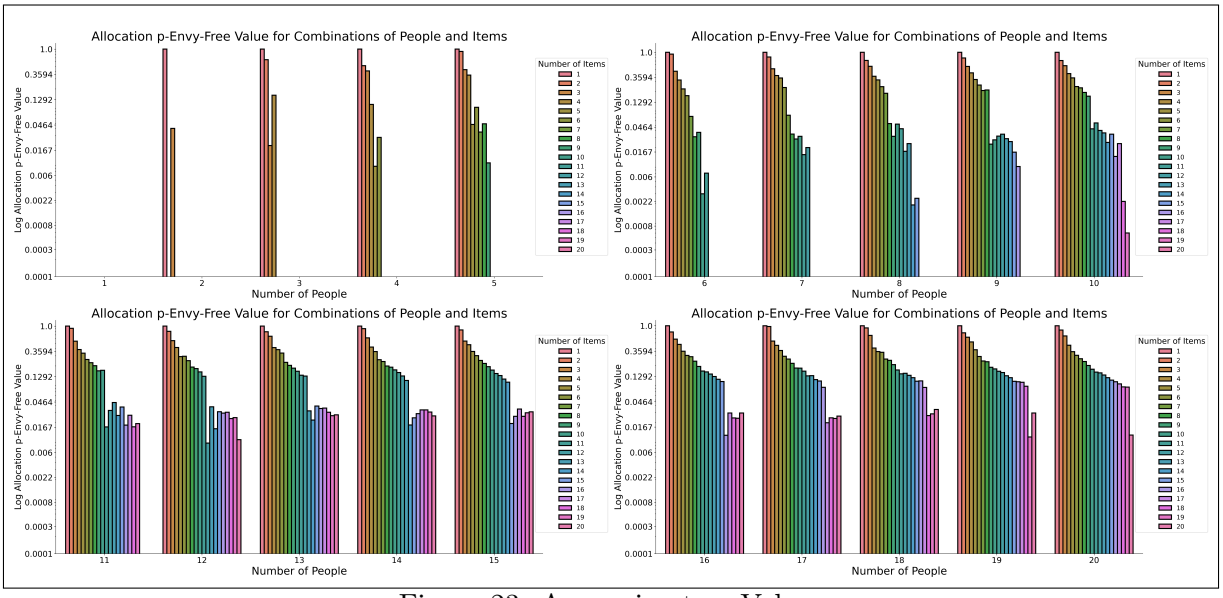


Figure 23: Approximate p Values

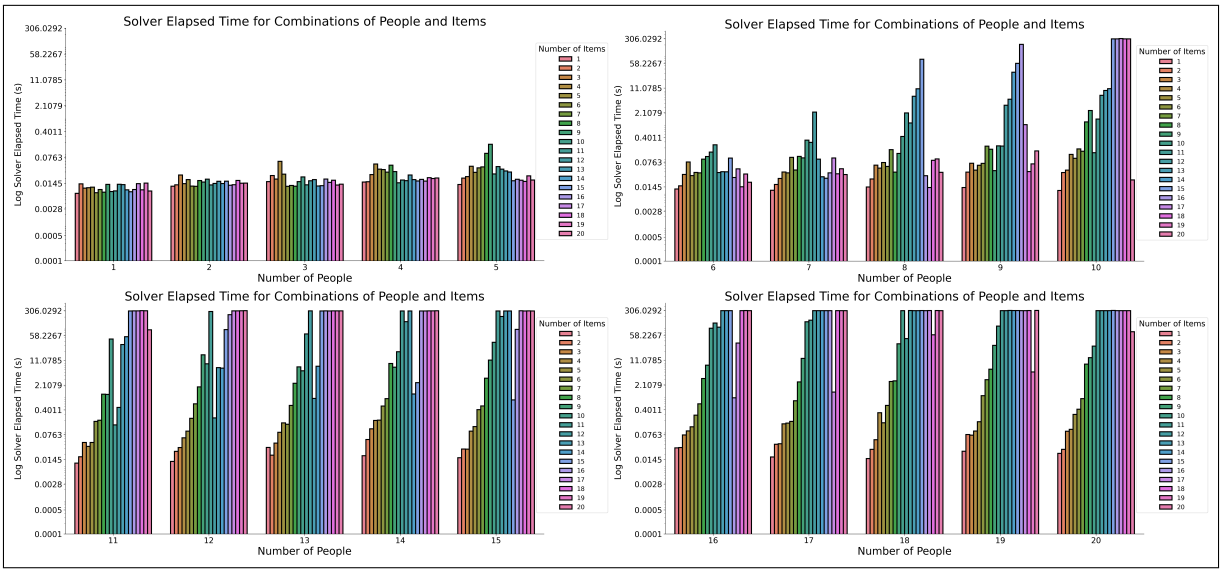
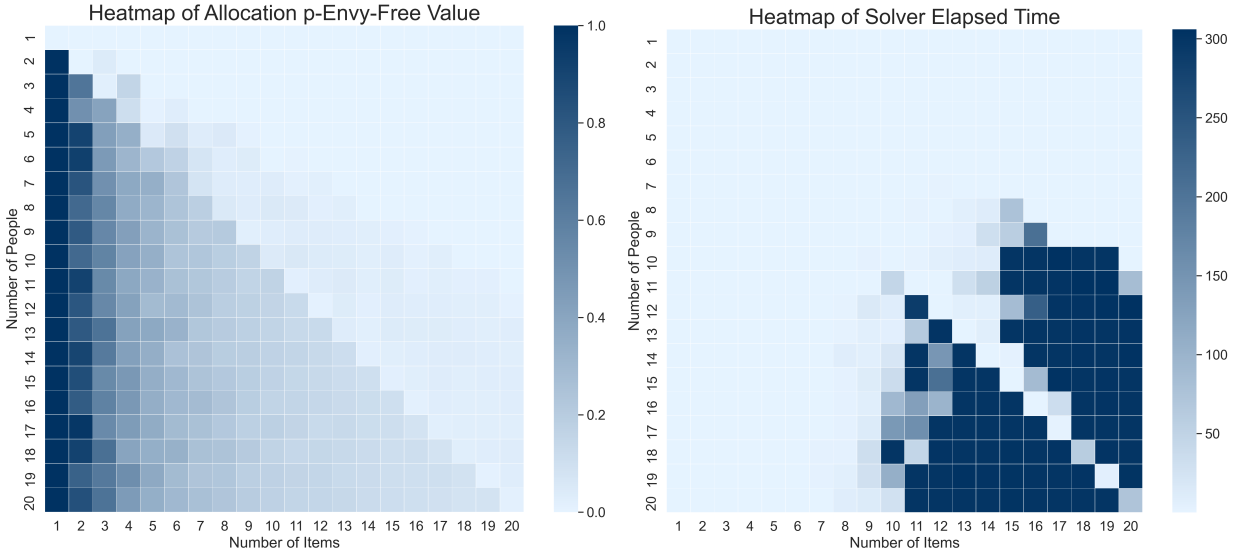
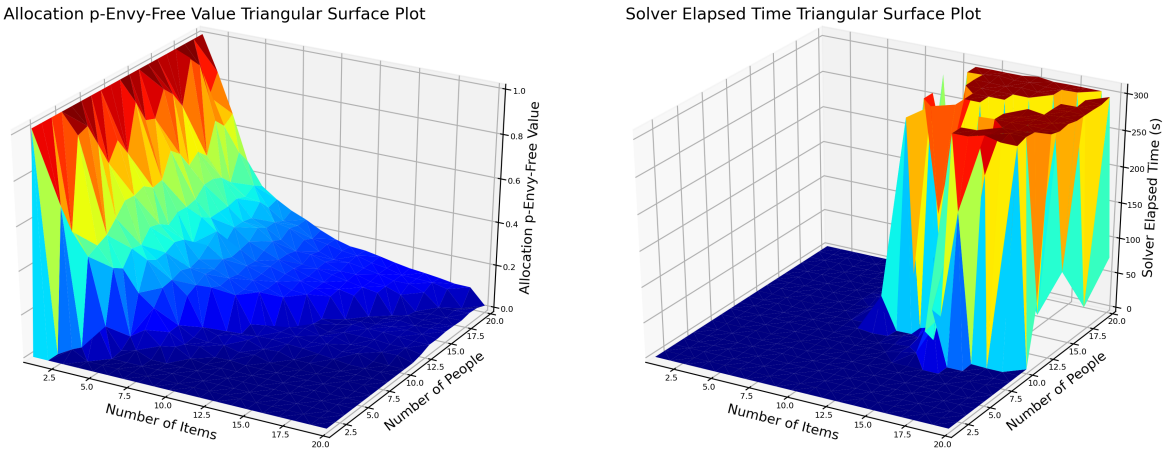


Figure 24: Solver Elapsed Time

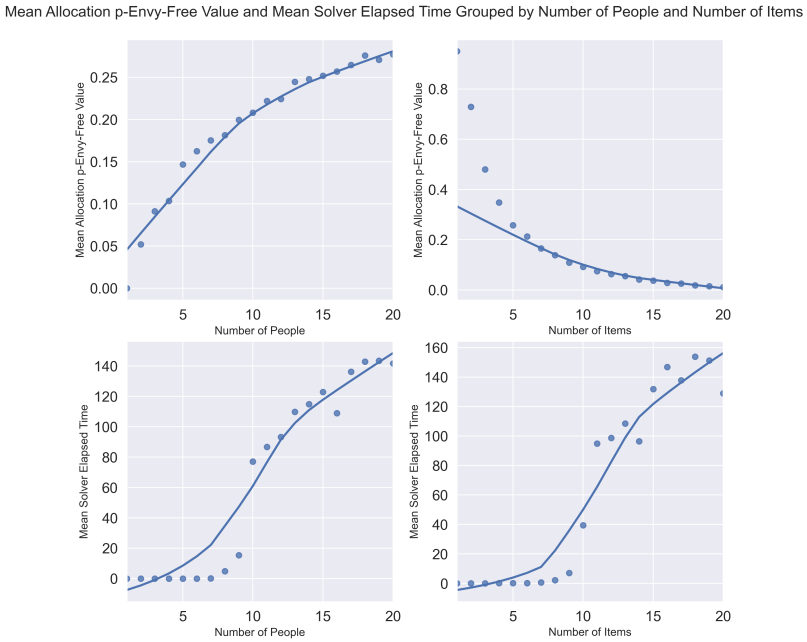
A.4.3.2 Heat Maps



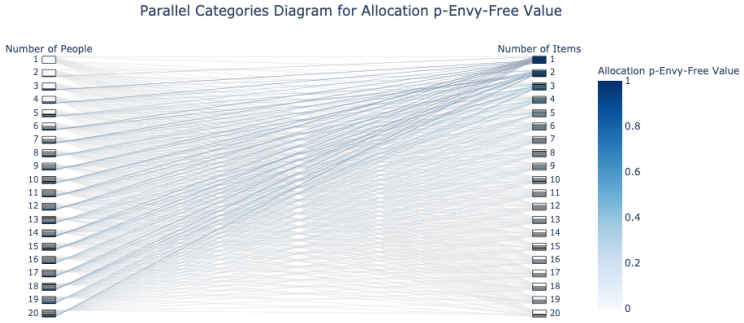
A.4.3.3 Triangular Surface Plots

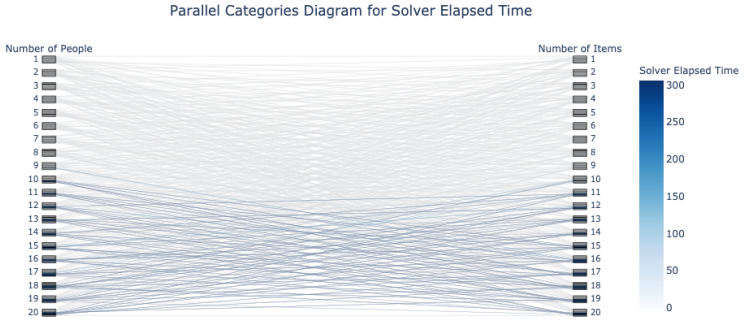


A.4.3.4 Scatter Plots of Mean Aggregated Data



A.4.3.5 Parallel Categories Diagrams





A.4.4 Solver Parameters Tuned

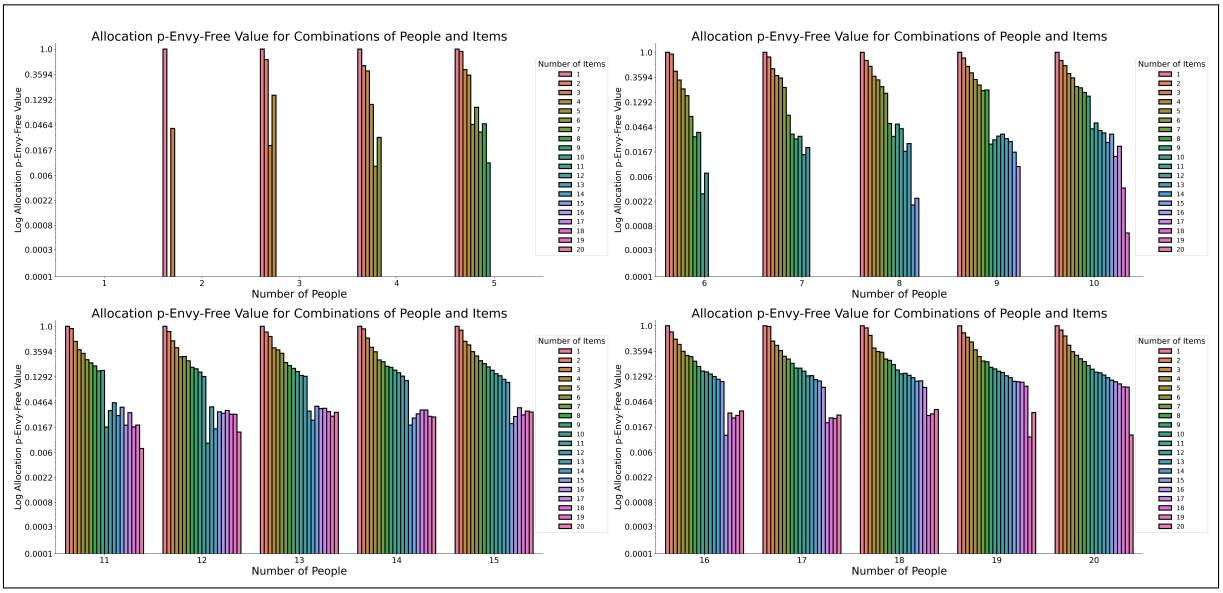


Figure 28: Approximate p Values

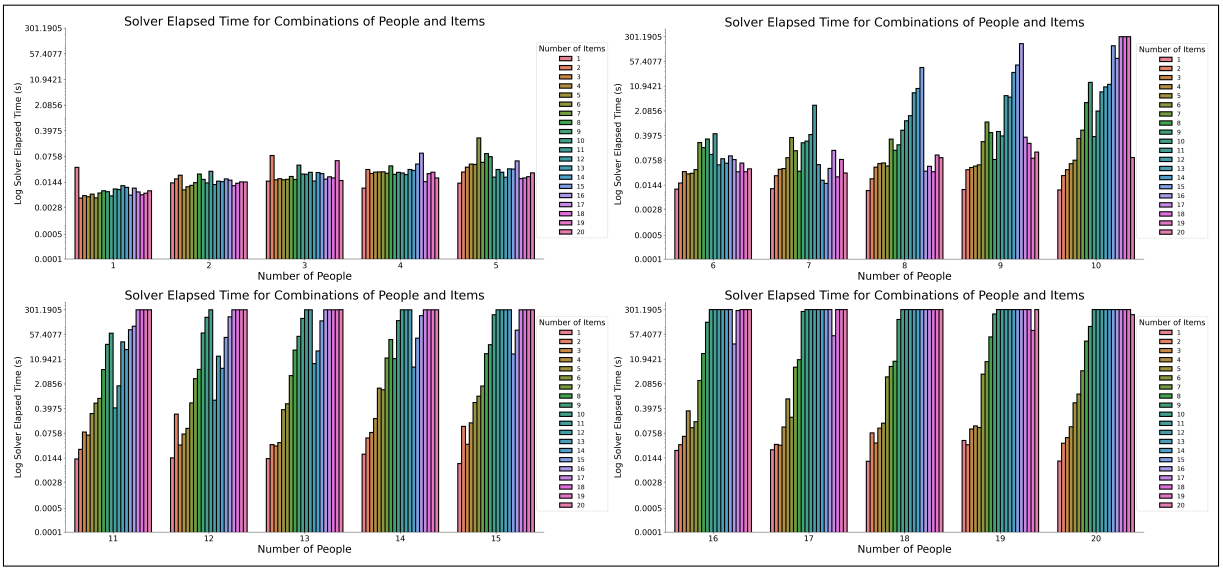
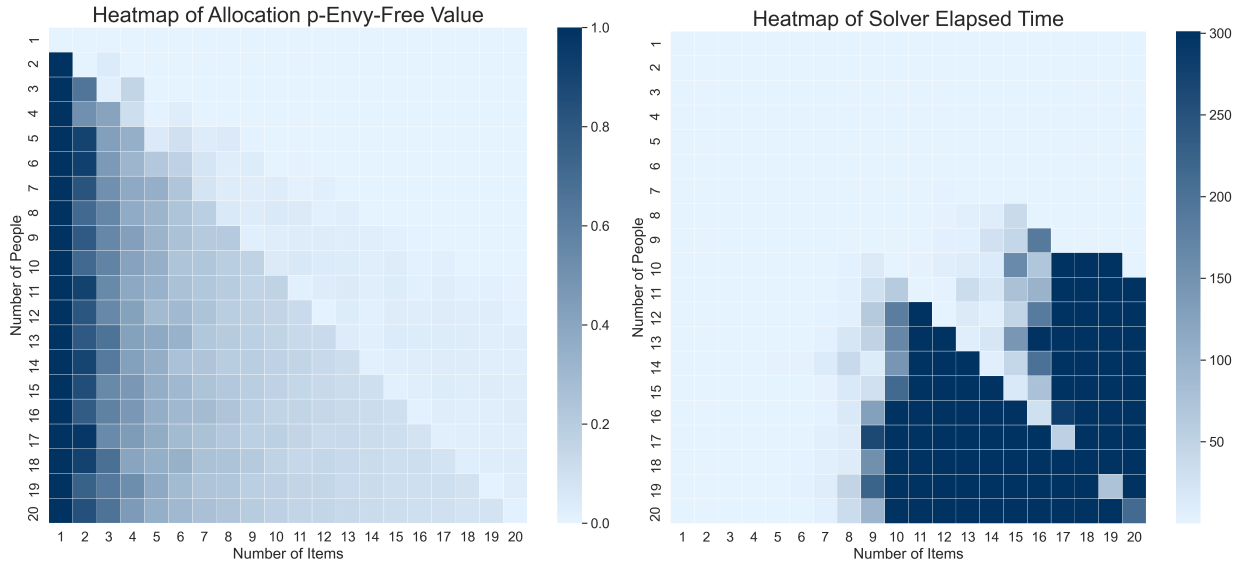
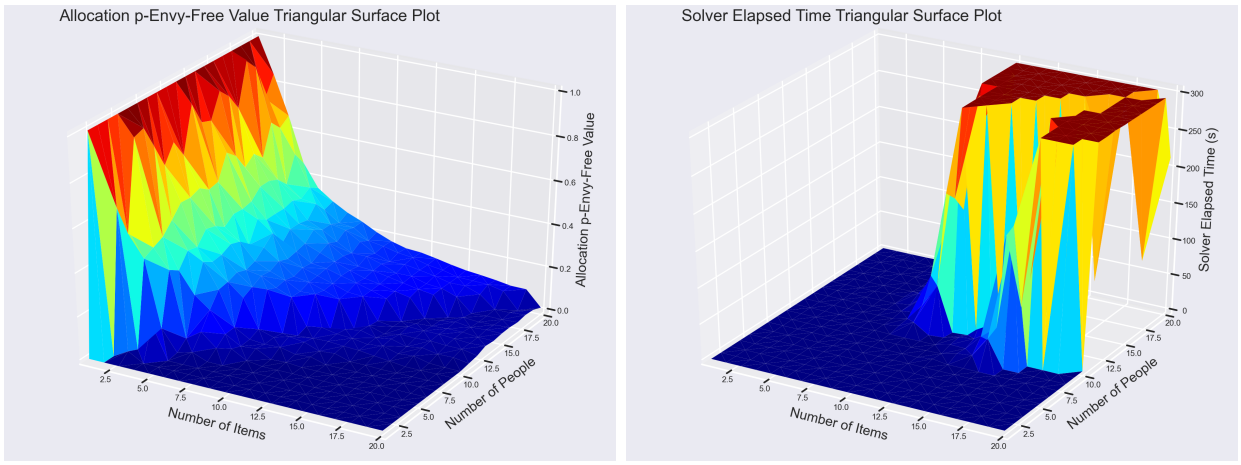


Figure 29: Solver Elapsed Time

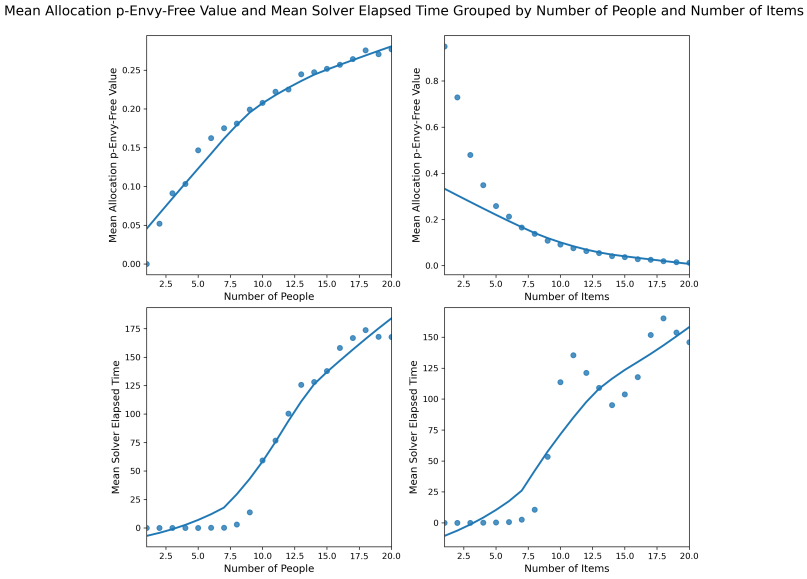
A.4.4.2 Heat Maps



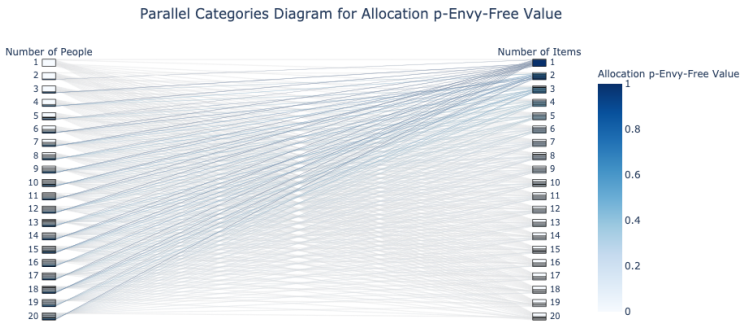
A.4.4.3 Triangular Surface Plots

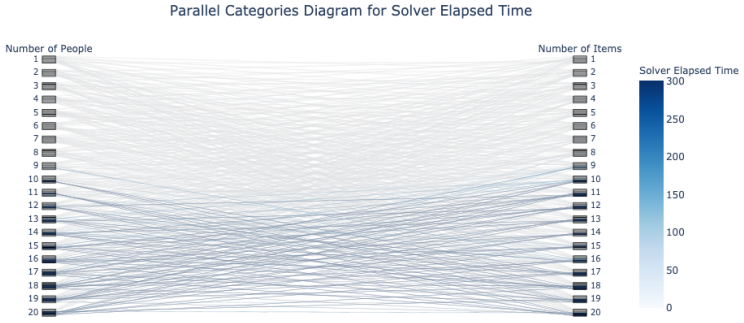


A.4.4.4 Scatter Plots of Mean Aggregated Data



A.4.4.5 Parallel Categories Diagrams





References

[1] Georgios Amanatidis, Georgios Birmpas, and Vangelis Markakis. "Comparing Approximate Relaxations of Envy-Freeness". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. Twenty-Seventh International Joint Conference on Artificial Intelligence {IJCAI-18}. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 42–48. ISBN: 978-0-9992411-2-7. DOI: 10.24963/ijcai.2018/6. URL: https://www.ijcai.org/proceedings/2018/6 (visited on 05/09/2020).

[2] Sarah Boxer. "For Birthday Parties or Legal Parties; Dividing Things Fairly Is Not Always a Piece of Cake". In: *The New York Times* (Aug. 7, 1999). ISSN: 0362-4331. URL: https://www.nytimes.com/1999/08/07/arts/for-birthday-parties-legal-parties-dividing-things-fairly-not-always-piece-cake.html (visited on 05/12/2020).

[3] Steven J. Brams, D. Marc Kilgour, and Christian Klamler. "Maximin Envy-Free Division of Indivisible Items". In: *Group Decision and Negotiation* 26.1 (Jan. 1, 2017), pp. 115–131. ISSN: 1572-9907. DOI: 10.1007/s10726-016-9502-x. URL: https://doi.org/10.1007/s10726-016-9502-x (visited on 05/11/2020).

[4] Johannes Brustle et al. "One Dollar Each Eliminates Envy". In: *arXiv:1912.02797 [cs, econ]* (Dec. 5, 2019). arXiv: 1912.02797. URL: http://arxiv.org/abs/1912.02797 (visited on 05/05/2020).

[5] Ioannis Caragiannis et al. "The Unreasonable Fairness of Maximum Nash Welfare". In: *Proceedings of the 2016 ACM Conference on Economics and Computation*. EC '16. Maastricht, The Netherlands: Association for Computing Machinery, July 21, 2016, pp. 305–322. ISBN: 978-1-4503-3936-0. DOI: 10.1145/2940716.2940726. URL: https://doi.org/10.1145/2940716.2940726 (visited on 05/02/2020).

[6] "CPLEX Parameters Reference". In: (), p. 190.

[7] *CPLEX Performance Tuning for Mixed Integer Programs*. Library Catalog: www.ibm.com. Aug. 22, 2016. URL: https://www.ibm.com/support/pages/cplex-performance-tuning-mixed-integer-programs (visited on 05/05/2020).

[8] John P Dickerson et al. "The Computational Rise and Fall of Fairness". In: (), p. 7.

[9] R. J. Lipton et al. "On approximately fair allocations of indivisible goods". In: *Proceedings of the 5th ACM conference on Electronic commerce*. EC '04. New York, NY, USA: Association for Computing Machinery, May 17, 2004, pp. 125–131. ISBN: 978-1-58113-771-2. DOI: 10.1145/988772.988792. URL: https://doi.org/10.1145/988772.988792 (visited on 04/30/2020).

[10] Andrea Lodi. "How to use your favorite MIP Solver: modeling, solving, cannibalizing". In: (), p. 99.

[11] Pasin Manurangsi and Warut Suksompong. "Asymptotic Existence of Fair Divisions for Groups". In: *Mathematical Social Sciences* 89 (Sept. 2017), pp. 100–108. ISSN: 01654896. DOI: 10.1016/j.mathsocsci.2017.05.006. arXiv: 1706.08219. URL: http://arxiv.org/abs/1706.08219 (visited on 05/12/2020).

[12] Pasin Manurangsi and Warut Suksompong. "When Do Envy-Free Allocations Exist?" In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.1 (July 17, 2019). Number: 01, pp. 2109–2116. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.33012109. URL: https://aaai.org/ojs/index.php/AAAI/article/view/4042 (visited on 05/12/2020).

[13] "Report of the Washington Meeting, September 6-18, 1947". In: *Econometrica* 16.1 (1948). Publisher: [Wiley, Econometric Society], pp. 101–106. ISSN: 0012-9682. URL: https://www.jstor.org/stable/1914289 (visited on 05/11/2020).