Kajibuntan: A House Chore Division App

Ayumi Igarashi and Tomohiko Yokoyama

The University of Tokyo
{igarashi,tomohiko_yokoyama}@mist.i.u-tokyo.ac.jp

Abstract

Couples often encounter the challenge of sharing house chores. This raises the fundamental question of how to divide chores. In this paper, we present a new application for a fair division of household chores. Our platform, called Kajibuntan, allows couples to specify the set of chores to be shared, their preferences over them, and the current allocation. Our tool visualizes the current allocation and makes proposals according to their preferences based on the theory of fair division. The goal of our tool is to provide a systematic and transparent system to divide household chores and help creating harmony in the home.

Introduction

Division of house chores can be a great source of conflicts. For example, some studies have reported that inequity in the division of household labor can cause significant stress to couples in the home (Bird 1999). A more recent study shows that unequal division causes women to consider breaking-up (Ruppanner, Brandén, and Turunen 2018). People may feel that their work is invisible and unacknowledged and therefore wish to have a constructive conversation to discuss a fair split of house chores.

This raises the fundamental question of how to divide chores. Is a mere 50:50 split of the total amount of work the perfect solution? The answer may be "no." The division of labor needs to fit people's lifestyles and interests. For example, some people find certain tasks, e.g., gardening, more bearable (or even joyful) than their partners do. Thus, the challenge lies in how to define fairness and achieve a fair allocation when people have complicated preferences.

Fair division is a growing field in mathematics and computer science that provides an elegant solution to this problem. It has been successfully implemented in various real-life applications, including the Adjusted Winner Website [1], Fair Division Calculator [2], Course Match [3] (Budish et al. 2017), and Spliddit [4] (Goldman and Procaccia 2015).

[1]http://www.nyu.edu/projects/adjustedwinner

[2]https://www.math.hmc.edu/ su/fairdivision/calc

[3]https://mba-inside.wharton.upenn.edu/course-match

[4]https://www.spliddit.org

In this paper, we present a new application of fair division to household chore. Our goal is to provide a systematic and transparent tool dividing household chores and promoting harmony in the home.

Model

A house chore division problem is a special case of a fair division problem of indivisible chores between two agents. We assume that there are two agents and m indivisible chores. Each agent $i \in \{1, 2\}$ has a disutility function $d_i \colon 2^{[m]} \to \mathbb{R}$ over the chores. We assume that the disutilities are additive, i.e., $d_i(S) = \sum_{j \in S} d_i(\{j\})$ for any subset S of chores. An *allocation* is a partition $A = (A_1, A_2)$ of the chores into two, where each A_i is the set of chores assigned to agent i.

Fairness and Efficiency Guarantees Envy-freeness is a central notion of fairness in the fair division literature. Formally, an allocation A is *envy-free* if no agent envies others, i.e., for any pair of agents i, j, we have $d_i(A_i) \leq d_i(A_j)$. However, when the resource to be allocated is indivisible, one cannot always guarantee the existence of an envy-free allocation. Therefore, the recent literature on fair division has extensively looked into an approximate fairness notion. Budish (2011) proposed an approximate notion of envy-freeness, called *envy-freeness up to one good* (EF1). In the context of chores allocation, an allocation is EF1 if the envy can be removed after the removal of one chore from an envious agent's bundle, i.e., for any pair of agents i, j, we have either $d_i(A_i) \leq d_i(A_j)$, or there exists a chore $c \in A_i$ such that $d_i(A_i \setminus \{c\}) \leq d_i(A_j)$ (Aziz et al. 2022).

In practice, we need to achieve both fairness and efficiency in resource allocation. Pareto-optimality is a commonly used notion of efficiency. An allocation A is called *Pareto-optimal* (PO) if there is no other allocation A' that strictly decreases at least one agent's disutility and does not increase the disutility of the other agents.

Discrete Adjusted Winner Algorithm The adjusted winner (AW) protocol is a classical fair division protocol for two agents dividing divisible items (Brams and Taylor 1996). Aziz et al. (2022) proposed a discrete version of the protocol that achieves Pareto-optimality and EF1 for indivisible chores. In our tool, we use the discrete adjusted winner algorithm to propose an alternative allocation. To illustrate the algorithm, let us call one agent a *winner* and an-

Figure 1: Input page

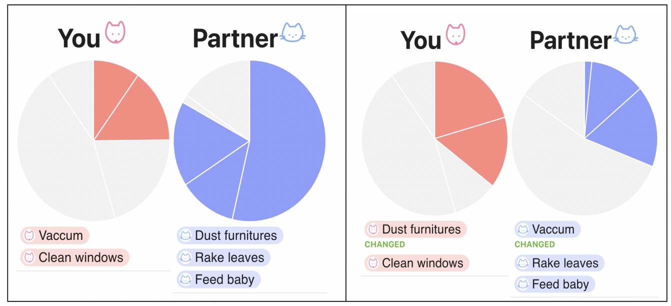

Figure 2: Output pages

other agent a *loser*. Initially, all chores are allocated to the loser, and as long as the loser's envy cannot be bounded up to one chore, each chore c is transferred from the loser to the winner step-by-step in the non-decreasing order of value $d_{\text{winner}}(\{c\})/d_{\text{loser}}(\{c\})$. Each step preserves Pareto-optimality. Hence, the final allocation is both EF1 and PO. We make the following modifications in the algorithm. First, because the discrete AW tends to allocate more to the loser, we decide who is the winner and who is the loser uniformly at random. Second, the modified version keeps transferring a chore from the winner to the loser even if the allocation already becomes EF1. We stop just before a new transfer of a chore causes the winner's envy or increases the difference between the disutilities (normalized to 1). In this way, we can bring our proposal closer to envy-freeness. Although the output does not necessarily satisfy envy-freeness, it is still an EF1 and Pareto-optimal allocation.

Main Functionalities

Now, we explain the main functionalities of our tool. The tool is available at:

http://housechore-division.com.

1. Enter Input To make visible all invisible work, our tool provides 52 items of daily house chores. Users select from predefined chores a set of chores that need to be shared with their partners. After the initial user tests, we discovered that it is often difficult for nonexperts to understand the concept of "disutilities." Thus, we disintegrate the meaning of the word into "time" and "preference." That is, we ask users to express both the time and preference—the amount of time that the users need to complete each chore and whether they are positive, neutral, or negative regarding each chore. The time is entered through a time slider and the preference is entered by clicking Like, Neutral, and Dislike icons. Figure 1 depicts the input page of our tool. The disutilities are then computed by multiplying time t and the number associated with each icon; specifically, each user assigns disutility t (respectively, $2t$ and $3t$) to a chore if they are positive (respectively, neutral and negative) regarding the chore.

2. Visualize the Current Allocation Next, our tool visualizes the current allocation. A pie chart represents the proportion of the chores that each user is in charge of according to the personal disutilities computed in the previous step; see the left of Figure 2. For example, if Alice's total disutility for all chores is 100 and she is in charge of cleaning, for which she has a disutility of 60, the colored region corresponds to the $3/5$ portion of the pie chart. Note that if one gets half of the pie at most, it means that the person does not envy their partner under the corresponding allocation.

3. Propose Alternative Allocations Finally, on the basis of the personal disutilities, the tool makes two types of proposals, that is, one that slightly modifies the current allocation by transferring one chore from an envious bundle to the envied bundle, and another that is produced by the modified discrete AW algorithm. The right of Figure 2 illustrates an example of a PO and EF1 allocation returned by the modified discrete AW. An additional feature is that users can exchange chores by clicking the corresponding chips, allowing them to customize the allocation to fit their personal constraints.

Implementation The implementation of the Kajibuntan web application is developed primarily by the React JavaScript library, and we use Next.js for deploying the web application and Netlify for hosting the web servers.

Social Impact and Discussion

The Japanese version of the tool was developed through an NHK TV program in Japan and broadcast on May 29, 2022.[5] Since the first month after launch, Kajibuntan has attracted almost $13,000$ unique visitors. We have received 131 answers to to our questionnaires. About 60% of them said that the tool is useful for them to discuss division of chores.

We believe that our work inspires several future directions. As in most social choice applications, we have found that eliciting agents' true preferences is a critical challenge. In particular, agents' preferences are often combinatorial, i.e., it is more efficient for users to combine a certain set of chores, such as cooking and buying groceries. More broadly, our tool has a potential to serve as a platform for fairness studies on AI techniques. For example, it would be interesting to empirically study what people believe to be "fair."

Acknowledgements

We thank a group of volunteers who worked on the Japanese version of the app at the social Hackathon organized by Code for Japan. This work was partially supported by JST PRESTO under grant number JPMJPR20C1.

[5]This version is available at http://kajibuntan.com.

References

Aziz, H.; Caragiannis, I.; Igarashi, A.; and Walsh, T. 2022. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1): 3:1–3:21.

Bird, C. E. 1999. Gender, household labor, and psychological distress: the impact of the amount and division of housework. *Journal of Health and Social Behavior*, 40(1): 32–45.

Brams, S. J.; and Taylor, A. D. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.

Budish, E. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6): 1061–1103.

Budish, E.; Cachon, G. P.; Kessler, J. B.; and Othman, A. 2017. Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation. *Operations Research*, 65(2): 314–336.

Goldman, J.; and Procaccia, A. D. 2015. Spliddit: Unleashing Fair Division Algorithms. *SIGecom Exchange*, 13(2): 41–46.

Ruppanner, L.; Brandén, M.; and Turunen, J. 2018. Does Unequal Housework Lead to Divorce? Evidence from Sweden. *Sociology*, 52(1): 75–94.