Fair Allocation of Conflicting Courses under Additive Utilities

Arpita Biswas
Harvard University
Cambridge, US
arpitabiswas@g.harvard.edu

Yiduo Ke
Northwestern University
Evanston, US
yiduo-ke@u.northwestern.edu

Samir Khuller
Northwestern University
Evanston, US
samir.khuller@northwestern.edu

Quanquan C. Liu
Yale University
New Haven, US
quanquan@mit.edu

ABSTRACT

We investigate the problem of fair allocation of indivisible items when certain item pairs conflict, where conflicts are represented by an interval graph. In this setting, no two conflicting items may be allocated to the same agent. Our problem has practical applications specifically for course allocation where students are agents and course seats are items; courses may have conflicting schedules. We devise algorithms for finding *fair*, specifically envy-freeness up to one item (EF1), allocations of courses to students in the most general setting: when students have non-uniform, non-identical, additive utility functions. In this extended abstract, we provide one of the algorithms that finds a EF1 solution under identical utilities, implying that, for any course, all students have the same utility.

KEYWORDS

Resource allocation, envy-freeness, course scheduling

ACM Reference Format:
Arpita Biswas, Yiduo Ke, Samir Khuller, and Quanquan C. Liu. 2024. Fair Allocation of Conflicting Courses under Additive Utilities. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Our work is motivated by the problem of allocating seats in courses in the university setting. Often, the demand greatly exceeds the capacity of these courses, limited by the availability of course staff, and so many students do not obtain their preferred courses. We seek a solution that every student deems *fair*.

In our problem setting, a student is an agent and each seat is an indivisible resource. We would ideally like to compute an *envy-free* assignment of seats to students. An *envy-free* allocation assigns seats to students such that no student envies another student's allocation. A student i envies another student i' if the sum of the utilities of the courses allocated to i is less than the sum of the utilities allocated to i' (using i's utility function); namely, student i envies i' when i has higher utility for the bundle of courses allocated to i'. Each seat occupies an interval of time corresponding to the course meeting time. Each student has non-negative utilities for each seat (course), and the total utility of a student is the sum of the utilities of the assigned courses calculated using that student's utility function. Having courses occupy intervals of time results in a very interesting and hitherto under-explored problem domain consisting of fair allocation of items *with conflicts*. Conflicts lead to a much harder problem setting; thus, fair solutions when no such conflicts exist could become infeasible allocations in our settings. Hence, our setting requires an entirely new set of techniques. Thus far, only a few recent papers have studied fair allocation with conflicting items [3, 6, 7, 10]. In this extended abstract, we propose a technique to obtain a fair solution under (non-binary) additive utilities, fixed credit caps, and strict restrictions on the allocation of conflicting courses. None of the existing solutions can simultaneously cater to all these challenges.

We now discuss the fairness criterion that we aim to achieve. While an assignment where no student gets a seat in any course is certainly envy-free (fair), unfortunately, such a solution is not desirable. As a result, we propose the concept of Charity to preclude such trivial solutions. All courses that remain unassigned at the end of running the course allocation algorithm become part of the Charity. Not only do we want students to envy other students as little as possible, we also want no student to envy the Charity. Specifically, no student should envy any non-conflicting set of courses in the Charity. This ensures non-trivial allocations.

Ideally, we desire an envy-free (EF) assignment of courses to students and also with the Charity. However, there are settings where such allocations are impossible. The impossibility of allocating indivisible items (such as course seats) without envy has led to defining a notion of *almost* envy-freeness, called *envy-freeness up to one item* (EF1) [5]. In our context, if a student i envies another student i', then there exists a course j' that has been allocated to student i' whose removal from the allocation of i' eliminates student i's envy towards i'. Although EF1 has been studied for capacity-constrained or knapsack settings [2, 4, 8, 9], these papers did not consider non-conflicting allocations, particularly pertinent in the context of course allocation. In this paper, we guarantee EF1 among students and envy-freeness with respect to Charity. We formally refer to this definition as EF1-CC+. Biswas et al. [3] introduced the specific course allocation problem we study in our paper and a weaker fairness notion, called EF1-CC. Although they introduced the problem setting, the results they present only hold for the very restrictive settings of binary and uniform utility functions. We significantly expand and extend their results into the much broader setting of additive utility functions.

For the first time, we establish that it is possible to find an EF1-CC+ allocation under non-identical additive utilities, when the credit caps are non-uniform. We present this result in the full version of our paper. In this extended abstract, we focus on the simpler setting of identical, additive utilities with uniform credit caps .

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 PRELIMINARIES

We consider a set of students, $[n]$, and a set of courses, $[m]$, where $[x]$ denotes the set of elements in $\{1, \ldots, x\}$ for any $x \in \mathbb{Z}^+$. The set of courses allocated to a student $i \in [n]$ is denoted by A_i. $\mathcal{A} = (A_1, \ldots, A_n)$ denotes an allocation. Courses are represented as intervals, with a start and end time. [1] The intervals that overlap are said to be in conflict with each other. Thus, the conflict graph of courses in this case is an *interval graph*.

Each course takes up one credit in the number of credits a student can take; each student can take at most k courses. In summary, our goal is to find an *fair* (defined below) allocation of at most k courses that do not overlap or conflict. Furthermore, no course can be assigned to more students than its seat capacity. We consider additive utility, generalizing the uniform and binary assumptions [1, 3, 10]. Each course j has an associated non-negative utility which is a non-negative integer $u_i(j) \in \{0, 1, \ldots, U\}$. The *maximum weighted independent set*, $\text{MWIS}^k(A_i)$ takes in a set of courses $[m]$ and outputs the maximum weighted set of size at most k non-conflicting courses with the weights given by uj for each course $j \in A_i$. Finding a maximum weighted independent set (MWIS) is polynomial time when the underlying graph is an interval graph.

Our fairness definition is a stronger version of the definition given by Biswas et al. [3] that considered only binary (i.e. {0, 1}) utilities. We extend these definitions to additive utilities. In addition, we consider stronger guarantees between any student and Charity (a dummy student who is assigned the set D of all unallocated courses/seats), namely that no student envies Charity. More formally, we define EF1-CC+ as a conflict-free allocation $\mathcal{A} = (A_1, \ldots, A_n)$,[2] where, for any pair of students $i, i' \in [n]$, either i does not envy i' or there exists at least one course $j' \in A_{i'}$ such that $u_i(A_i) \geq u_i(\text{MWIS}^k(A_i) \setminus \{j'\})$. $\text{MWIS}^k(A_i)$ is the maximum weight independent set of size at most k in $A_{i'}$ using u_i as the weights. Moreover, for all students $i \in [n]$, it holds that $u(A_i) \geq u(S_{c_i})$ for all $S_{c_i} \in \mathcal{S}$, where $\mathcal{S}$ contains all independent sets of size at most k in the Charity; in other words, *no* student envies Charity (the set of unassigned courses).

3 ENVY-FREE COURSE ALLOCATION CONSIDERING CHARITY

We devise Algorithm 1 that returns an EF1-CC+ allocation for n students with identical, additive, non-negative, integral utilities. The algorithm has two phases. After the first phase ends, the allocation satisfies EF1 among all the students, and the end of the second phase ensures EF1-CC+.

In the first phase, each course is sorted by non-decreasing end time in Line 2, A_i are empty sets, and D is initialized to $[m]$. According to the sorted order, a course j is offered to the student, say i, with the current lowest utility $u(A_i)$. Then, we compare the utility of the current allocation $u(A_i)$ to the maximum weighted independent set of size $\leq k$ of the union between her current allocation and j, $(A_i \setminus Conf(A_i, j)) \cup \{j\}$, where $Conf(A_i, j)$ is the set of courses in A_i that conflict with j. Then, i accepts the maximum of the two independent sets A_i and $(\{j\} \cup A_i \setminus Conf(A_i, j))$ (Line 8). The courses assigned to i are removed from the D and any rejected courses from A_i end up in the Charity D.

In the second phase of the algorithm, we assign courses from Charity by making increasingly larger offers to the student with the current lowest utility (Line 15). We offer the first course (in earliest end time order by Line 12) of the Charity to the student with the lowest utility. If the value of the largest independent set of the union between this course and the student's own allocation is at most the value of her current allocation, then the student declines the offer (Line 23). After declining, we then offer the bundle of the first two courses in the sorted order, and if that also doesn't suffice, then offer the first three, and so on. We denote the first t courses in D by $D_{[t]}$. If the student accepts the offer (Line 16) and takes the maximum independent set of size $\leq k$ (Line 17), then the leftover courses go to the Charity (Line 18). After an offer is accepted, this process starts over again with the earliest-ending course in the Charity and the current lowest utility student (Line 21). The algorithm terminates when the lowest utility student declines the offer of the entire Charity.

Algorithm 1 EF1-CC+ for Identical Additive Utilities and Uniform Credit Caps: Greedy and Gradual Improve algorithm

Input: Students $[n]$, courses $[m]$, identical credit cap k and identical additive non-negative utility function u.
Output: EF1-CC+ allocation $\mathcal{A}$.

1: **Phase 1:**
2: $M \leftarrow [m]$; Sort M by earliest end time
3: Initialize all $A_i = \emptyset$
4: Initialize $D \leftarrow [m]$ ▷ All courses go to Charity
5: **for** course j in M **do**
6: Let $i = \arg\min_{s \in [n]} (u(A_s))$ ▷ Lowest utility student
7: **if** $\text{MWIS}^k(A_i \cup j) > u(A_i)$ **then**
8: $A'_i = \text{MWIS}^k(A_i \cup j)$
9: $D \leftarrow (D \cup A_i) \setminus A'_i$ ▷ Update the Charity
10: $A_i \leftarrow A'_i$ ▷ Update the allocation of i
11: **Phase 2:**
12: Sort and maintain D dynamically by earliest end time
13: Let $i = \arg\min_{s \in [n]} u(A_s)$ ▷ Lowest utility student
14: $t = 1$
15: **while** $t \leq |D|$ **do**
16: **if** $\text{MWIS}^k\left(D_{[t]} \cup A_i\right) > u(A_i)$ **then**
17: $A'_i = \text{MWIS}^k\left(D_{[t]} \cup A_i\right)$
18: $D \leftarrow (D \cup A_i) \setminus A'_i$ ▷ Update the Charity
19: $A_i \leftarrow A'_i$ ▷ Update the allocation of i
20: t = 1
21: $i = \arg\min_{s \in [n]} u(A_s)$ ▷ Lowest utility student
22: **else**
23: $t \leftarrow t + 1$
24: Return $\mathcal{A} = \{A_1, \ldots, A_n\}$

Theorem 1. *Algorithm 1 outputs an EF1-CC+ allocation under identical additive non-negative utilities and uniform credit caps in* $O\left(n \cdot m \cdot \sum_{j=1}^{m} u(j)\right)$ *time.*

[1] In this model, we assume a course meets once a week but, in practice, typically, many universities have courses meeting on Mon-Wed-Fri, Mon-Wed, or Tu-Th during the same time period on each of those days. If we pick the Mon-Tue schedule, we can essentially capture the schedule for almost all courses.

[2] A *conflict-free* allocation is one where no two classes in any bundle A_i overlap.

Proof. For establishing EF1-CC+, we need to prove (1) by the end of the algorithm, no student envies another student by more than one course and (2) no student envies Charity.

To prove the second claim is easy: the algorithm terminates when the lowest-utility student no longer desires to swap any courses from Charity. In other words, she does not envy the Charity. Since all the students have identical utilities and uniform credit caps, the other students (with utilities at least that of the lowest-utility student) also do not envy the Charity.

We prove the first claim: for any two students s and s', student s never envies s' by more than one course throughout the algorithm. We prove this claim for Phase 1 and Phase 2 separately. We first prove via contradiction that when Phase 1 ends, our allocation satisfies EF1. Suppose that student s envies s' by more than one item. Then it is easy to see that the last course given to s' should not have been offered to s' but someone else, as s' was not the lowest utility student before its last course was assigned.

Now we prove this claim for Phase 2 via induction. The base case: the first step of the algorithm assigns the first course to the lowest-utility student, which could be any student, since everyone starts out with an empty allocation. Thus, if s' is assigned the first course, then s only envies s' by at most one course. For the inductive step, we order the sets of courses given to students by the algorithm; let this order be σ. For example, if courses $\{x, y, z\}$ is the i-th set of courses to be given simultaneously to a student, say student s_j, then $\sigma_i = \{x, y, z\}$; if the next round of courses to be given to a student consists of only course q, then $\sigma_{i+1} = \{q\}$. We say "set" because in the second phase of the algorithm (from Line 12 to Line 24), multiple courses from the Charity can be assigned simultaneously to a student. Assuming the inductive hypothesis where s does not envy s' by more than one course after the set of courses σ_i is assigned, we have to prove that s does not envy s' by more than one course after σ_{i+1} is assigned.

In Phase 2, if s envies s' by one course after σ_i, then that means s' has a higher utility than s, and thus s' will not be assigned the set of courses σ_{i+1}, which is the same argument as above. If s did not envy s' after the assignment of courses in σ_i, however, we must make a different argument than above due to the fact that multiple courses can be assigned simultaneously to a student in the second phase of the algorithm. If s' is the lowest-utility student after the assignment of σ_i and gets assigned σ_{i+1} from the Charity, then s would envy s' by at most one course. The reason for this is due to the selection of the value t in Line 16. Since t is the smallest number such that letting s' make a new bundle for herself by including the first t courses in the Charity makes s' strictly prefer that new bundle to her current allocation $A_{s'}$. On the other hand, by Lines 17 and 23, a bundle including just the first $t - 1$ courses from Charity and the current allocation of s' is valued at strictly lower than $A_{s'}$. This means that if s were to envy the new allocation of s' after the assignment of σ_{i+1}, we can remove D_t from that σ_{i+1} so that s no longer envies s'. The claim holds for σ_l where l is the index of the last set of courses assigned, and any two students are symmetrical. This completes the proof. □

ACKNOWLEDGEMENTS

Y. Ke, S. Khuller, and Q. C. Liu gratefully acknowledge support from NSF-Award 2216970 (IDEAL Institute) and Northwestern startup.

REFERENCES

[1] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. 2021. Finding fair and efficient allocations for matroid rank valuations. *ACM Transactions on Economics and Computation* 9, 4 (2021), 1–41.

[2] Arpita Biswas and Siddharth Barman. 2018. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 91–97.

[3] Arpita Biswas, Yiduo Ke, Samir Khuller, and Quanquan C. Liu. 2023. An Algorithmic Approach to Address Course Enrollment Challenges. In *4th Symp. on Foundations of Responsible Computing, FORC 2023, June 7-9, 2023, Stanford University, California, USA (LIPIcs, Vol. 256)*, Kunal Talwar (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 8:1–8:23. https://doi.org/10.4230/LIPIcs.FORC.2023.8

[4] Arpita Biswas, Justin Payan, Rik Sengupta, and Vignesh Viswanathan. 2023. *The Theory of Fair Allocation Under Structured Set Constraints, Ethics in Artificial Intelligence: Bias, Fairness and Beyond.* Springer, Singapore, 115–129. https://doi.org/10.1007/978-981-99-7184-8_7

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

[6] Nina Chiarelli, Matjaž Krnc, Martin Milanič, Ulrich Pferschy, Nevena Pivač, and Joachim Schauer. 2022. Fair Allocation of Indivisible Items with Conflict Graphs. *Algorithmica* (2022), 1–31.

[7] Halvard Hummel and Magnus Lie Hetland. 2022. Fair allocation of conflicting items. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 8.

[8] Hila Shoshan, Noam Hazon, and Erel Segal-Halevi. 2023. Efficient Nearly-Fair Division with Capacity Constraints. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 206–214.

[9] Warut Suksompong. 2021. Constraints in fair division. *ACM SIGecom Exchanges* 19, 2 (2021), 46–61.

[10] Vignesh Viswanathan and Yair Zick. 2023. Yankee Swap: A Fast and Simple Fair Allocation Mechanism for Matroid Rank Valuations. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 179–187.