An Online Allocation Algorithm of Indivisible Goods

Kohei Shimizu and Yoshifumi Manabe
Department of Computer Science
Kogakuin University
Tokyo, Japan
e-mail: em15010@ns.kogakuin.ac.jp , manabe@cc.kogakuin.ac.jp

***Abstract*—This paper proposes a new online allocation algorithm of indivisible goods. In online algorithms, participants arrive to execute the algorithm at any time and exit from the algorithm when his/her allocation is given. We assume that the total value of the whole goods is the same for every participant. In cake-cutting algorithms for divisible goods, immediately envy-free has been defined as the desirable property. The property means that for any participants, any other participants who arrive after and depart before the participant obtain no more value than the participant. However, it is difficult for online allocation algorithms for indivisible goods to satisfy immediately envy-free. Therefore we propose a weakly immediately envy-free algorithm, which means that participants do not value goods allocated to participants who arrived later but departs earlier than them, more than their own. Our algorithm aims to maximize the worst obtained value among all participants. We show that this problem involves an NP-complete problem. Thus, it is very difficult to always output an optimal solution. We propose an approximation algorithm and prove its approximation ratio.**

Keywords-allocation; envy-free; indivisible goods; algorithm

I. INTRODUCTION

We consider the problem of fair allocation of multiple indivisible goods among multiple participants. This paper discusses an online allocation problem of indivisible goods that models allocating multiple presents among multiple participants in a party, paying rewards to multiple employee by indivisible goods, and so on. We aim to satisfy weakly immediately envy free, which means that participants do not value goods allocated to the participants who arrived later but departs earlier than them, more than their own. In addition, our algorithm aims to maximize the worst participant's obtained value of allocated goods among the participants. Several approximation algorithms have been discussed for different rating scales, such as the maximin share guarantee [1-3]. However, these algorithms first divide the goods into groups such that the minimum of each group's total value is guaranteed, and then assign the groups to the participants to maximize the sum of each participant's obtained value. Maximizing the sum is the aim of the second phase of the algorithm, and these algorithms are offline allocation algorithm. For divisible goods, online cake cutting algorithm was proposed [4], but as far as the authors know, there is no online algorithm of indivisible goods. We prove that this problem contains a case when solving an NP-complete problem is necessary. Thus we propose an approximation algorithm and prove its approximation ratio. By a computer simulation, we show that the approximation ratio of our algorithm is much better than the worst case bound.

II. PROBLEM DEFINITION

This paper discusses an online allocation problem of indivisible goods. The allocation problem is defined as follows.

The set of participants is denoted by $X = \{x_1, x_2, ..., x_n\}$.

The set of goods is denoted by $V = \{v_1, v_2, ..., v_m\}$.

The evaluation function of each participant for the goods is denoted by $P_i = (1 \le i \le n) : V \to \mathrm{N}$.

We assume that the evaluation functions satisfies the following:

$$\forall i, j (1 \le i \le n, 1 \le j \le m) P_i(v_j) \ge 1,$$

$$\textstyle\sum_{j=1}^{m} P_i(v_j) = P (1 \le i \le n) .$$

That is, the total evaluation of the whole goods is the same for every participant. Though the actual evaluation values might differ among the participants, they are normalized. This assumption is natural for the allocation problem, since obtaining all goods is the best result and the ratio compared with the best result is each participant's interest. In this paper we call the unit of evaluation as a point. In addition, there is no good that has no value for any participants. We assume that all participants are risk-adverse.

An allocation is function $A : X \to 2^V$ (Subset of V).

It must satisfy the following equations: $\bigcup_{i=1}^{n} A(x_i) = V$ and $A(x_i) \cap A(x_{i'}) = \phi$ for any $i, i' (i \ne i')$.

The total points of allocated goods for x_i by allocation algorithm A is denoted by $u_i(A) = \sum_{v_j \in A(x_i)} P_i(v_j)$.

The minimum value of u_i by the allocation A is denoted by $u(A) = \min_{x_i \in X} u_i(A)$.

The optimal $u(A)$ by an exhaustive search that ignores weakly immediately envy-free is denoted by $u(\widetilde{A})$.

This paper considers that the best allocation is as follows. (1) The allocation must be weakly immediately envy free, which means that participants do not value goods allocated to participants who arrived later but departs earlier than them, more than their own. (2) If there are multiple allocations that satisfy (1), $u(A)$ is the largest.

In many allocation problems, the participant who departs the allocation before the next participant arrives tends to cause bad effect for the allocation [4]. As a penalty, if there is a participant who wants to depart the allocation before the next participant arrives, we do not consider about weakly immediately envy free for the participant.

III. NP-COMPLETENESS

Even no matter how the participants arrive, there is an instance when the number of participants who have not yet been allocated is two. This allocation problem contains a case when solving an NP-complete problem is necessary. When the last allocation is executed between the last two participants, obtaining an optimal allocation is NP-complete, as shown in the following, because it belongs to NP and it has a reduction from a partition problem.

For the proof of NP-completeness, let us consider the following decision problem [5-7].

Input: X, V, P_i, and integer k

Question: Is there A such that $u(A) \geq k$?

It is obvious that this decision problem belongs to NP. NP-hardness can be proved by a reduction from the following decision problem of the partition problem.

Input: Set of integers $\chi = \{s_1, s_2, ..., s_p\}$, that satisfies

$$\sum_{s_i \in \chi} s_i = 2L$$

Question: Is there an allocation (χ_1, χ_2) such that $\chi = \chi_1 \cup \chi_2$, $\chi_1 \cap \chi_2 = \phi$, and $\sum_{s_i \in \chi_1} s_i = \sum_{s_i \in \chi_2} s_i$?

When $n = 2, m = p, P_1 = P_2$ and $k = L$, the partition problem has a solution if and only if the allocation problem has a solution. For example, partition problem ($\chi = \{3,8,4,12,5,9,1,2,2,6\}$, $2L = 52$) can be converted to an allocation problem ($n = 2$, $m = 10$, $P = 52$, $P_1 = \{3,8,4,12,5,9,1,2,2,6\}$, $P_2 = \{3,8,4,12,5,9,1,2,2,6\}$, $k = 2L/2 = 26$). In this case, there is allocation A, $A(x_1) = \{v_1, v_3, v_4, v_7, v_{10}\}$, $A(x_2) = \{v_2, v_5, v_6, v_8, v_9\}$, $u(A) = 26$, as $\chi_1 = \{3,4,12,1,6\}$, $\chi_1 = \{8,5,9,2,2\}$.

It is obvious that the partition problem has a solution if and only if the corresponding allocation problem has a solution.

Since the partition problem is NP-complete, the allocation problem contains a case when solving an NP-complete problem is necessary. Because of the NP-completeness, we consider an approximate solution that is as close as possible to the optimal solution.

IV. THE PROPOSED ALGORITHM

We propose an approximation algorithm. Note that participants have P_i that satisfies the condition in the problem definition.

1) Wait for participants until the number of participants who are waiting for allocation is two or more. If a participant wants to depart the allocation before the next participant arrives, he/she can get any one good he/she prefers and depart.

2) Every participant checks currently remaining goods, and declares the minimum number of goods they are satisfied. (Every participant cannot declare more than T, where T is the number of remaining goods divided by the number of participants who have not been obtained his/her allocation.)

3) From the declared numbers, the participant who declares the minimum number gets the number of goods whatever he/she likes. If there are multiple participants who declare the minimum number, the participant who arrived the earliest gets the number of goods whatever he/she likes. The participant who got goods departs from the allocation.

4) Repeat step 1, 2 and 3 until the number of participants who have not yet been allocated is one.

5) The last participant gets all the remaining goods and departs.

The proposed algorithm tends to keep more goods for participants who have not yet arrived. We cannot know P_i of a player who has not yet arrived, and there are no goods whose value is 0, therefore to keep more goods tend to lead a good allocation. Since all participants are risk-adverse, participants do not declare false number.

If the proposed algorithm is not weakly immediately envy-free, there is a participant who knows another participant's $A(x_i)$ and feel envy. Because of the characteristics of step 2 and 3 of the proposed algorithm, the participant who arrives earlier than the remaining participants can definitely get more value than he/she is satisfied. If he declares the minimum number, he can get goods that he wants, and if another participant declares a smaller number than his one, the participant gets low-value goods for him and departs. Therefore, this algorithm outputs weakly immediately envy-free solution.

Let us discuss the early depart rule in step 1. Consider the case when one participant is waiting for next participant. If the participant declares one in the next round, the participant definitely wins in the round and obtains one good he/she prefers because of the rule in step 3. Thus, the participant can depart without waiting for the next participant if he/she is satisfied with one good.

V. EXAMPLE

A. Example 1

Input: $n=3$, $m=4$, $P=100$, $P_1=\{30,30,20,20\}$, $P_2=\{80,10,5,5\}$, $P_3=\{30,10,20,40\}$.

The optimal solution for this example, that is obtained by an exhaustive search, is as follows. $A(x_1)=\{v_2,v_3\}$, $A(x_2)=\{v_1\}$, $A(x_3)=\{v_4\}$, $u_1(A)=50$, $u_2(A)=80$, $u_3(A)=40$, $u(\widetilde{A})=40$.

Suppose the following order of arrivals. The first participant x_1 and the second participant x_2 arrive. The number of participants who have not yet been allocated is three. The sum of the value of remaining goods is 100 for x_1, thus x_1 is satisfied if he gets at least $\frac{100}{3}$ points. The sum of the value of remaining goods is 100 for x_2, thus x_2 is satisfied if he gets at least $\frac{100}{3}$ points. x_1 declares that x_1 is satisfied with two goods, for example, $A(x_1)=\{v_1,v_2\}$. x_2 declares that x_2 is satisfied with one good, for example, $A(x_2)=\{v_1\}$. x_2 declares a smaller number than x_1, thus x_2 can get one good whatever he wants and depart. x_2 gets v_1 and departs. The third participant x_3 arrives after x_2's depart. Currently, the number of participants who have not yet been allocated is two. The sum of the value of remaining goods is 70 for x_1, thus x_1 is satisfied if he gets at least $\frac{70}{2}$ points. The sum of the value of remaining goods is 70 for x_3, thus x_3 is satisfied if he gets at least $\frac{70}{2}$ points. x_1 declares that x_1 is satisfied with two goods, for example, $A(x_1)=\{v_2,v_3\}$. x_3 declares that x_3 is satisfied with one good, for example, $A(x_3)=\{v_4\}$. x_3 declares a smaller number than x_1, thus x_3 can get one good whatever he wants and depart. x_3 gets v_4 and departs. Currently, the number of participants who have not yet been allocated is one. x_1 gets all remaining goods and departs. x_1 gets v_2 and v_3, and departs.

In this case, $A(x_1)=\{v_2,v_3\}$, $A(x_2)=\{v_1\}$, $A(x_3)=\{v_4\}$, $u_1(A)=50$, $u_2(A)=80$, $u_3(A)=40$, $u(A)=40$. In this example, the proposed algorithm outputs the optimal solution.

B. Example 2

Input: $n=4$, $m=8$, $P=100$, $P_1=\{50,20,2,2,2,2,2,2\}$, $P_2=\{10,20,20,10,10,10,10,10\}$, $P_3=\{20,20,10,10,10,10,10,10\}$, $P_4=\{5,5,5,5,10,20,30,20\}$.

The optimal solution for this example, that is obtained by an exhaustive search, is as follows. $A(x_1)=\{v_1\}$, $A(x_2)=\{v_2,v_3\}$, $A(x_3)=\{v_4,v_5,v_8\}$, $A(x_4)=\{v_6,v_7\}$, $u_1(A)=50$, $u_2(A)=40$, $u_3(A)=30$, $u_4(A)=50$, $u(\widetilde{A})=30$.

Suppose the following order of arrivals. The first participant x_1 and the second participant x_2 arrive. The number of participants who have not yet been allocated is four. The sum of the value of remaining goods is 100 for x_1, thus x_1 is satisfied if he gets at least $\frac{100}{4}$ points. The sum of the value of remaining goods is 100 for x_2, thus x_2 is satisfied if he gets at least $\frac{100}{4}$ points. x_1 declares that x_1 is satisfied with one good, for example, $A(x_1)=\{v_1\}$. x_2 declares that x_2 is satisfied with two goods, for example, $A(x_2)=\{v_2,v_3\}$. x_1 declares a smaller number than x_2, thus x_1 can get one good whatever he wants and depart. x_1 gets v_1 and departs. The third participant x_3 arrives after x_1's depart. Currently, the number of participants who have not yet been allocated is three. The sum of the value of remaining goods is 90 for x_2, thus x_2 is satisfied if he gets at least $\frac{90}{3}$ points. The sum of the value of remaining goods is 80 for x_3, thus x_3 is satisfied if he gets at least $\frac{80}{3}$ points. x_2 declares that x_2 is satisfied with two goods, for example, $A(x_2)=\{v_2,v_3\}$. x_3 declares that x_3 is satisfied with two goods, for example, $A(x_3)=\{v_2,v_3\}$. x_2 and x_3 declares the same number, thus the earliest participant x_2 can get two goods whatever he wants and depart. x_2 gets v_2 and v_3, and departs. The fourth participant x_4 arrives after x_2's depart. Currently, the number of participants who have not yet been allocated is two. The sum of the value of remaining goods is 50 for x_3, thus x_3 is satisfied if he gets at least $\frac{50}{2}$ points. The sum of the value of remaining goods is 85 for x_4, thus x_4 is satisfied if he gets at least $\frac{85}{2}$ points. x_3 declares that x_3 is satisfied with three goods, for example, $A(x_3)=\{v_4,v_5,v_6\}$. x_4 declares that x_4 is satisfied with two goods, for example, $A(x_4)=\{v_6,v_7\}$. x_4 declares a smaller number than x_3, thus x_4 can get two goods whatever he wants and depart. x_4 gets v_6 and v_7, and departs. Currently, the number of participants who have not yet been allocated is one. x_3 gets all remaining goods and departs. x_3 gets v_4, v_5, and v_8, and departs.

In this case, $A(x_1)=\{v_1\}$, $A(x_2)=\{v_2,v_3\}$, $A(x_3)=\{v_4,v_5,v_8\}$, $A(x_4)=\{v_6,v_7\}$, $u_1(A)=50$, $u_2(A)=40$, $u_3(A)=30$, $u_4(A)=50$, $u(A)=30$. In this case, x_3 knows $A(x_4)$, x_4 arrives later than x_3 and departs before than x_3. If x_3 feels envy to x_4, the allocation is not weakly immediately envy-free. However, $u_3(A)=30$ and $u_3(A(x_4))=20$. x_3 does not feel envy to x_4.

In this example, the proposed algorithm outputs the weakly immediately envy-free and optimal solution.

C. Example 3

Input: $n = 4$, $m = 8$, $P = 100$, $P_1 = \{20,20,15,10,10,10,10,5\}$, $P_2 = \{15,15,15,15,15,15,5,5\}$, $P_3 = \{2,3,5,10,15,15,25,25\}$, $P_4 = \{35,10,10,10,10,1010,5\}$.

The optimal solution for this example, that is obtained by an exhaustive search, is as follows. $A(x_1) = \{v_2, v_3\}$, $A(x_2) = \{v_4, v_5, v_6\}$, $A(x_3) = \{v_7, v_8\}$, $A(x_4) = \{v_1\}$, $u_1(A) = 35$, $u_2(A) = 45$, $u_3(A) = 50$, $u_4(A) = 35$, $u(\widetilde{A}) = 35$.

Suppose the following order of arrivals. The first participant x_1 and the second participant x_2 arrive at this allocation. The number of participants who have not yet been allocated is four. The sum of the value of remaining goods is 100 for x_1, thus x_1 is satisfied if he gets at least $\frac{100}{4}$ points. The sum of the value of remaining goods is 100 for x_2, thus x_2 is satisfied if he gets $\frac{100}{4}$ points. x_1 declares that x_1 is satisfied with two goods, for example, $A(x_1) = \{v_1, v_2\}$. x_2 declares that x_2 is satisfied with two goods, for example, $A(x_2) = \{v_1, v_2\}$. x_1 and x_2 declares the same number, thus the earliest participant x_1 can get two goods whatever he wants and depart. x_1 gets v_1 and v_2, and departs. The third participant x_3 arrives after x_1 's depart. Currently, the number of participants who have not yet been allocated is three. The sum of the value of remaining goods is 70 for x_2, thus x_2 is satisfied if he gets at least $\frac{70}{3}$ points. The sum of the value of remaining goods is 95 for x_3, thus x_3 is satisfied if he gets at least $\frac{95}{3}$ points. x_2 declares that x_2 is satisfied with two goods, for example, $A(x_2) = \{v_3, v_4\}$. x_3 declares that x_3 is satisfied with two goods, for example, $A(x_3) = \{v_7, v_8\}$. x_2 and x_3 declares the same number, thus the earliest participant x_2 can get two goods whatever he wants and depart. x_2 gets v_3 and v_4, and departs. The fourth participant x_4 arrives after x_2 's depart. Currently, the number of participants who have not yet been allocated is two. The sum of the value of remaining goods is 80 for x_3, thus x_3 is satisfied if he gets at least $\frac{80}{2}$ points. The sum of the value of remaining goods is 35 for x_4, thus x_4 is satisfied if he gets at least $\frac{35}{2}$ points. x_3 declares that x_3 is satisfied with two goods, for example, $A(x_3) = \{v_7, v_8\}$. x_4 declares that x_4 is satisfied with two goods, for example, $A(x_4) = \{v_5, v_6\}$. x_3 and x_4 declares the same number, thus the earliest participant x_3 can get two goods whatever he wants and depart. x_3 gets v_7 and v_8, and departs. Currently, the number of participants who have not yet been allocated is one. x_4 gets all remaining goods and departs. x_4 gets v_5 and v_6, and departs.

In this case, $A(x_1) = \{v_1, v_2\}$, $A(x_2) = \{v_3, v_4\}$, $A(x_3) = \{v_7, v_8\}$, $A(x_4) = \{v_5, v_6\}$, $u_1(A) = 40$, $u_2(A) = 30$, $u_3(A) = 50$, $u_4(A) = 20$, $u(A) = 20$. The proposed algorithm does not output the optimal solution for this example. The reason is that we do not know P_4 until x_4 arrives. But this allocation is weakly immediately envy free.

VI. Approximation Ratio

In this paper, we define the approximation ratio as the ratio of $u(A)$ by the proposed algorithm to $u(\widetilde{A})$ ($u(A)$ by an offline exhaustive search). The approximation ratio is proved for the three cases, $n = m$, $2n \le m$ and $n+1 \le m \le 2n-1$.

A. $n = m$

If $n = m$, every participant declares that he/she is satisfied with one good. If there is a participant who cannot get his/her most valuable good, $u(\widetilde{A}) \le \frac{P}{2}$. If every participant values different good as the best one for him/her, they can get the good they want by the proposed algorithm, because of the characteristics of step 2 and 3 of the proposed algorithm. If there are participants who values the same good as the best one, at least one participant cannot obtain his/her best good, thus $u(\widetilde{A}) \le \frac{P}{2}$. In this case, $u(A) \ge 1$, because each participant can get one good and there is no good that has no value for any participants. Therefore, in the case of $n = m$, the approximation ratio is at least $\frac{1}{\frac{P}{2}} = \frac{2}{P}$.

B. $2n \le m$

In this algorithm, there is an upper limit T for the declare number. (Every participant cannot declare more than T, where T is the number of remaining goods divided by the number of participants who have not been obtained his/her allocation.) Every participant can get more than one goods such that he/she is satisfied with. Therefore, every participant can get at least 2 points. $u(\widetilde{A})$ is less than P points. Therefore, in $2n \le m$, the approximation ratio is at least $\frac{1}{\frac{P}{2}} = \frac{2}{P}$.

C. $n+1 \le m \le 2n-1$

In this algorithm, a participant who arrives the earliest at the time can at least get goods that he/she is satisfied. If $u(A) = 1$, there is a participant who gets only one good. In that case, there is a round that two participants declare two and someone definitely gets two goods. If a participant declare two, each value of all the remaining goods for the participant is less than $\frac{P}{n'}$ points, where n' is the number of participants who have not been obtained his/her allocation at the time. When $n' = 2$, there are three cases. The first case is when the number of the remaining goods is two. In this case, there is a round that two participants declare two and gets two goods. In this case, one of the participants gets two goods, less than $\frac{P}{n'}$ points, and $3 \leq n'$ at that round. $u(\widetilde{A})$ is maximized $\frac{P}{n'} \times 2 = \frac{2P}{3}$. The second case is when the number of the remaining goods is three. If $n+1=m$, there is no round that two participants declare two and someone gets two goods. In this case, if $u(A)$ by the proposed algorithm is 1, the last participant and another participant need the same good as the most valuable good. In this case, $u(\widetilde{A}) \leq \frac{P}{2}$. If $n+2 \leq m$, there is a round that two participants declare two, $u(\widetilde{A})$ is less than $\frac{2P}{3}$. The third case is when the number of the remaining goods more than three, $u(A)$ by the proposed algorithm is 2 and more. If there is not a round that two participants declare two, as mentioned in $n=m$, participants can get the good they want. In all cases, $u(\widetilde{A}) \leq P$. If $u(A)$ is 2 and more, the approximation ratio is at least $\frac{2}{P}$. Therefore, in the case of $n+1=m$, the approximation ratio is at least $\frac{1}{\frac{P}{2}} = \frac{2}{P}$. In the case of $n+2 \leq m$, the approximation ratio is at least $\frac{1}{\frac{2P}{3}} = \frac{3}{2P}$.

VII. Experiment Results

This algorithm is executed for 1000 randomly generated problem instances with $n = 3,4,$ and 5, $m = 3,4,5,6,7,8,9,$ and 10, and $P = 100$. The table shows the average ratio of $u(A)$ by the proposed algorithm to $u(\widetilde{A})$.

In this table, the proposed algorithm is compared with an exhaustive search that ignores weakly immediately envy-free. With that in mind, the proposed algorithm can output good solutions. The ratio tends to decrease if n increases, because the exhaustive search ignores weakly immediately envy-free.

TABLE I. Experiment Results

	n=3	n=4	n=5
m=3	0.7498	None	None
m=4	0.7440	0.5716	None
m=5	0.7494	0.5611	0.4146
m=6	0.7177	0.5623	0.4187
m=7	0.7340	0.5534	0.4279
m=8	0.7203	0.5520	0.4470
m=9	0.7188	0.5830	0.4972
m=10	0.7375	0.6013	0.5145

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP26330019.

References

[1] Haris, A., Serge, G., Simon, M., Toby, W. : "Fair Assignment of Indivisible Objects under Ordinal Preference", AAMAS '14, 2014, pp. 1305-1312.

[2] Walsh,T. : "Online Cake Cutting", Algorithmic Decision Theory, COMSOC 2010, pp 292-305.

[3] Jonathan, G., Ariel, D. P. : "Spliddit: Unleashing Fair Division Algorithms", ACM SIGecom Exchange, Vol.13, No.2, 2014, pp. 41-46.

[4] Procaccia, A. D., Wang, J. : "Fair enough: Guaranteeing approximate maximin shares", 14th ACM Conference on Economics and Computation, 2014, pp. 675-692.

[5] Sylvain, B., Michel, L. : "Characterizing Conflicts in Fait Division of Indivisible Goods Using a Scale of Criteria", AAMAS '14, 2014, pp. 1321-1328.

[6] Richard, L., Evangelos, M., Elchanan, M., Amin, S. : "On approximately fair allocations of indivisible goods", 5th ACM conference on Electronic commerce, 2004, pp. 125-131.

[7] Shimizu,K., Manabe,Y. : "An Allocation Algorithm of Indivisible Goods", 10th APSITT, August 2015, pp 112-114.