An Allocation Algorithm of Indivisible Goods

Kohei Shimizu and Yoshifumi Manabe
Email: em15010@ns.kogakuin.ac.jp, manabe@cc.kogakuin.ac.jp
Department of Informatics
Kogakuin University
1-24-2, Nishi-Shinjuku, Shinjuku, Tokyo, 163-8677 Japan

***Abstract*—This paper proposes a new allocation algorithm of indivisible goods. We consider the case when the total value of the whole goods is the same for every participant, which models allocation at divorce or inheritance. The worst participant's obtained value must be maximized. There are not good allocation algorithms for our rating scale. We show that this problem is NP-complete. Therefore we propose four types of approximation algorithms. Among the four algorithms, the raising standard algorithm has the best ratio that the algorithm outputs the optimal solution by a computer simulation.**

I. Introduction

We consider the problem of fair allocation of multiple indivisible goods among multiple participants. We aim 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][2][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. Since maximizing the sum is the aim of the second phase of the algorithm, the algorithm cannot obtain a good allocation result by the above criteria. We prove that this problem is NP-complete. Thus we propose several approximation algorithms. By a computer simulation, we show that the raising standard algorithm gives good allocation results.

II. Problem definition

This paper discusses an allocation problem of indivisible goods that models the property division at the time of divorce or inheritance. The allocation problem is defined as follows.

The set of participants is denoted by $X = \{x_1, x_2, \ldots, x_n\}$.
The set of goods is denoted by $V = \{v_1, v_2, \ldots, v_m\}$.
The evaluation function of each participant for the goods is denoted by $P_i (1 \leq i \leq n) : V \rightarrow \mathbb{N}$.
We assume that the evaluation functions satisfies the following: $\forall i, j (1 \leq i \leq n, 1 \leq j \leq m) P_i(v_j) \geq 1$, $\sum_{j=1}^{m} P_i(v_j) = P (1 \leq i \leq n)$,
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 at a divorce, 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.
An allocation is function $A : X \rightarrow 2^V$ (Subset of V).
It must satisfy the following equations: $\cup_{i=1}^{n} A(x_i) = V$ and $A(x_i) \cap A(x_{i'}) = \emptyset$ for any $i, i' (i \neq i')$.
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 sum of each participant's obtained points by the allocation A is denoted by $U(A) = \sum_{i=1}^{n} u_i(A)$.
The number of participants who obtained $u(A)$ points by the allocation A is denoted by $N(u(A))$.

This paper considers that the best allocation is as follows. (1) $u(A)$ is the largest, (2) if there are multiple allocations that satisfies (1), $N(u(A))$ is the smallest among them, (3) if there are multiple allocations that satisfies (2), $U(A)$ is the largest among them.

III. NP-completeness

This allocation problem is NP-complete for $n \geq 2$, as shown in the following, because it belongs to NP and it has a reduction from a partition problem[4][5].

For the proof of NP-completeness, let us consider the following decision problem.

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 $\chi = \{s_1, s_2, \ldots, s_p\}$, that satisfies $\sum_{s_i \in \chi} s_i = 2L$
Question: Is there an allocation (χ_1, χ_2) such that $\chi = \chi_1 \cup \chi_2, \chi_1 \cap \chi_2 = \emptyset$, 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 $\chi_1 = \{3, 4, 12, 1, 6\}$, $\chi_2 = \{8, 5, 9, 2, 2\}$.
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 is also NP-complete. Because of the NP-

completeness, we consider an approximate solution that is as close as possible to the optimal solution.

IV. The proposed algorithms

We propose several approximation algorithms that are compared in a later section. In any algorithm, first of all, participants declare P_i that satisfies the condition in the problem definition, to the allocation manager. The allocation manager decides the allocation using the functions. In this section, the set of currently assigned goods is denoted by $B(x_i)$(Initially, $B(x_i) = \emptyset$ for every x_i).

A. Maximum point priority algorithm

This algorithm tends to assign each good to the participant who values the good highest. Though this is the simplest algorithm, it leads a good allocation especially in the cases with two participants.

[Procedure]

1) Choose the next assignment subjects S ($S = \{x_i \in X|\ u_i(B)$ is the smallest$\}$) as the set of participants who obtained the fewest points.
2) Choose one of the highest point goods v ($v \in \mathrm{argmax}_{v_j \in V} \max_{x_i \in S} P_i(v_j)$). Assign v to x, one of the participants who values v highest ($x \in \mathrm{argmax}_{x_i \in S} P_i(v)$), and $V \leftarrow V - \{v\}$.
3) Repeat step 1 and 2 until all the goods are assigned.

B. Point difference priority algorithm

This algorithm tends to assign each good to the participant, such that the other participants do not want it so much. This characteristics is considered to lead an allocation such that the smallest objection occurs for the assignment of each good from the unassigned participants. Therefore, this algorithm outputs a good allocation in the cases that there is participant who wants the good that the other participants do not want.

[Procedure]

1) Same as step 1 in maximum point priority algorithm.
2) Choose one of the goods v ($v \in \mathrm{argmax}_{v_j \in V} \max_{x_i, x_k \in S}(P_i(v_j) - P_k(v_j))$) such that the difference between the largest evaluation value and the second evaluation value is the largest. Assign v to x, one of the participants who values v highest ($x \in \mathrm{argmax}_{x_i \in S} P_i(v)$), and $V \leftarrow V - \{v\}$. If there is only one participant in S ($S = \{x_i\}$), Assign v ($v \in \mathrm{argmax}_{v_j \in V} P_i(v_j)$) to x_i, and $V \leftarrow V - \{v\}$.
3) Repeat step 1 and 2 until all the goods are assigned.

C. Raising standard algorithm

This algorithm tends to assign each good to the participant who has many points of goods that were already assigned to another participant. This characteristics is considered to avoid the worst result that a participant obtains very few points. This algorithm especially works well in the cases with more than three participants.

[Procedure]

1) Choose the next assignment subjects S ($S = \{x_i \in X|\ u_i(B)$ is the smallest$\}$) as the set of participants who obtained the fewest points.
2) Choose one of the highest point goods v ($v \in \mathrm{argmax}_{v_j \in V} \max_{x_i \in S} P_i(v_j)$). Assign v to x, one of the participants who values v highest ($x \in \mathrm{argmax}_{x_i \in S} P_i(v)$), and $V \leftarrow V - \{v\}$.
3) For every participants $x_i \neq x$, and every unassigned good v_j, $P_i(v)$ is added to $P_i(v_j)$. The modified P_i is used for the further assignment in step 2.（Use the original P_i in step 1, and the final evaluation.）
4) Repeat step 1, 2, and 3 until all the goods are assigned.

D. Average consideration raising standard algorithm

In raising standard algorithm, there are cases when the algorithm lost the optimal solution at the time to choose the first assignment good. To avoid such cases, this algorithm is different from the raising standard algorithm in choosing the highest average point good among the remaining goods in the step 2. Therefore, there are cases that raising standard algorithm can not output the optimal solution but this algorithm can.

[Procedure]

1) Same as step 1 in raising standard algorithm.
2) Choose one of the highest average point goods v ($v \in \mathrm{argmax}_{v_j \in V}(\sum_{x_i \in S} P_i(v_j)/|S|)$). Assign v to x, one of the participants who values v highest ($x \in \mathrm{argmax}_{x_i \in S} P_i(v)$), and $V \leftarrow V - \{v\}$.
3) Same as step 3 in raising standard algorithm.
4) Repeat step 1, 2, and 3 until all the goods are assigned.

V. Example

Input: $n = 3,\ m = 6,\ P = 100,\ P_1 = \{50, 1, 29, 9, 7, 4\}$, $P_2 = \{3, 42, 44, 7, 3, 1\}, P_3 = \{30, 20, 42, 5, 2, 1\}$.
The optimal solution for this example, that is obtained by an exhaustive search, is as follows. $A(x_1) = \{v_1\}$, $A(x_2) = \{v_2, v_5, v_6\}$, $A(x_3) = \{v_3, v_4\}$, $u_1(A) = 50$, $u_2(A) = 46$, $u_3(A) = 47$, $u(A) = 46$, $U(A) = 143$, $N(u(A)) = 1$.

A. Maximum Point priority algorithm

The first assignment subjects are x_1, x_2 and x_3, and the maximum point among the remaining goods is $P_1(v_1) = 50$, so v_1 is assigned to x_1. Next assignment subjects are x_2 and x_3, and the maximum point among the remaining goods is $P_2(v_3) = 44$, so v_3 is assigned to x_2. Next assignment subject is x_3, and the maximum point among the remaining goods is $P_3(v_2) = 20$, so v_2 is assigned to x_3. By repeating the same procedure, v_4 is assigned to x_3, v_5 is assigned to x_3 and v_6 is assigned to x_3.

In this case, $A(x_1) = \{v_1\}$, $A(x_2) = \{v_3\}$, $A(x_3) = \{v_2, v_4, v_5, v_6\}$, $u_1(A) = 50$, $u_2(A) = 44$, $u_3(A) = 28$, $u(A) = 28$, $U(A) = 122$, $N(u(A)) = 1$.

B. Point difference priority algorithm

The first assignment subjects are x_1, x_2 and x_3, and the greatest difference between the 1st evaluation value and the 2nd evaluation value among the remaining goods is 22 for the good v_2 since the 1st evaluation value is 42 by x_2 and the 2nd evaluation value is 20 by x_3. Therefore, v_2 is assigned to x_2. Next assignment subjects are x_1 and x_3, and the greatest difference between the 1st evaluation value and the 2nd evaluation value among the remaining goods is 20 for the good v_1 by x_1, so v_1 is assigned to x_1. Next assignment subject is x_3, and the highest point good among the remaining goods is v_3, so v_3 is assigned to x_3. Next assignment subjects are x_2 and x_3, and the next assignment good is v_4, so v_4 is assigned to x_2. By repeating the same procedure, v_5 is assigned to x_3 and v_6 is assigned to x_3.

In this case, $A(x_1) = \{v_1\}$, $A(x_2) = \{v_2, v_4\}$, $A(x_3) = \{v_3, v_5, v_6\}$, $u_1(A) = 50$, $u_2(A) = 51$, $u_3(A) = 45$, $u(A) = 45$, $U(A) = 146$, $N(u(A)) = 1$.

C. Raising standard algorithm

The first assignment subjects are x_1, x_2 and x_3, and the maximum point good among the remaining goods is $P_1(v_1) = 50$, so we assign v_1 to x_1. $P_2(v_1) = 3$ and $P_3(v_1) = 30$, so 3 points are added to $P_2(v_i)(i = 2, 3, 4, 5,$ and 6) and 30 points are added to $P_3(v_i)(i = 2, 3, 4, 5,$ and 6). Therefore, we will use the following modified evaluation value $P_2 = \{3, 45, 47, 10, 6, 4\}$ and $P_3 = \{30, 50, 72, 35, 32, 31\}$ in the step 2 of the next round. Next assignment subjects are x_2 and x_3, and the maximum point good among the remaining goods is $P_3(v_3) = 72$, so v_3 is assigned to x_3. $P_1(v_3) = 29$ and $P_2(v_3) = 44$, so we will use the following modified evaluation value $P_1 = \{79, 30, 29, 38, 36, 33\}$ and $P_2 = \{47, 89, 47, 54, 50, 48\}$ in the step 2 of the next round. Next assignment subject is x_2, and the maximum point good among the remaining goods is $P_2(v_2) = 89$, so v_2 is assigned to x_2. We will use the following modified evaluation value $P_1 = \{80, 30, 30, 39, 37, 34\}$ and $P_3 = \{50, 50, 92, 55, 52, 51\}$ in the step 2 of the next round. By repeating the same procedure, v_4 is assigned to x_3, v_5 is assigned to x_2 and v_6 is assigned to x_2

In this case, $A(x_1) = \{v_1\}$, $A(x_2) = \{v_2, v_5, v_6\}$, $A(x_3) = \{v_3, v_4\}$, $u_1(A) = 50$, $u_2(A) = 46$, $u_3(A) = 47$, $u(A) = 46$, $U(A) = 143$, $N(u(A)) = 1$.

D. Average consideration raising standard algorithm

The first assignment subjects are x_1, x_2 and x_3, and the maximum average point good among the remaining goods is v_3 as $\frac{115}{3}$, so v_3 is assigned to x_2. $P_1(v_3) = 29$ and $P_3(v_3) = 42$, so 29 points are added to $P_1(v_i)(i = 1, 2, 4, 5,$ and 6) and 42 points are added to $P_3(v_i)(i = 1, 2, 4, 5,$ and 6). Therefore we will use the following modified evaluation value $P_1 = \{79, 30, 29, 38, 36, 33\}$ and $P_3 = \{72, 62, 42, 47, 44, 43\}$ in the step 2 of the next round. Next assignment subjects are x_1 and x_3, and the maximum average point good among the remaining goods is v_1 as $\frac{151}{2}$, so v_1 is assigned to x_1. $P_2(v_1) = 3$ and $P_3(v_1) = 30$, so we will use the following modified evaluation value $P_2 = \{3, 45, 47, 10, 6, 4\}$ and $P_3 = \{72, 92, 72, 77, 74, 73\}$ in the step 2 of the next round. Next assignment subject is x_3, and the maximum average point good among the remaining goods is v_2, so v_2 is assigned to x_3. We will use the following modified evaluation value $P_1 = \{80, 30, 30, 39, 37, 34\}$, $P_2 = \{45, 45, 89, 52, 48, 46\}$ in the step 2 of the next round. By repeating the same procedure, v_4 is assigned to x_3, v_5 is assigned to x_3 and v_6 is assigned to x_3.

In this case, $A(x_1) = \{v_1\}$, $A(x_2) = \{v_3\}$, $A(x_3) = \{v_2, v_4, v_5, v_6\}$, $u_1(A) = 50$, $u_2(A) = 44$, $u_3(A) = 28$, $u(A) = 28$, $U(A) = 122$, $N(u(A)) = 1$.

VI. RESULTS

These algorithms are executed for 2000 randomly generated problem instances with $n = 2, 3, 4$, and 5, $m = 5, 6, 7, 8, 9$, and 10 and $P = 100$. The number of times when the solution with the largest $u(A)$ solution is obtained and the number of times when the optimal solution is obtained by each algorithm are shown in the following table 1 and 2.

TABLE I. EXPERIMENTAL RESULT 1

Algorithms	Largest $u(A)$ solution is obtained
Maximum point algorithm	1087
Point difference priority algorithm	1240
Raising standard algorithm	1579
Average consideration raising standard algorithm	1332

TABLE II. EXPERIMENTAL RESULT 2

Algorithms	Optimal solution is obtained
Maximum point algorithm	1024
Point difference priority algorithm	1169
Raising standard algorithm	1325
Average consideration raising standard algorithm	1064

Among the four algorithms, the raising standard algorithm has the best ratio that the algorithm outputs the optimal solution. We consider the character of the raising standard algorithm fits the highest priority issue in this paper. Therefore we propose the raising standard algorithm is the best approximation algorithm. Bounding the approximation ratio is a future challenge.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number 26330019.

REFERENCES

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

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

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

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

[5] 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.