ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

TEORÍA DE DECISIÓN EN LA ASIGNACIÓN DE RECURSOS PARA LOGRAR EL BIENESTAR SOCIAL

MODELO PARA LA ASIGNACIÓN DE BIENES Y TAREAS INDIVISIBLES

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO MATEMÁTICO

HENRY CRISTHIAN BELTRÁN AVEIGA

henry.beltran@epn.edu.ec

DIRECTOR: IGSIL AUGUSTO DÁVILA MONTENEGRO

igsil.davila@epn.edu.ec

DMQ, AGOSTO 2022

CERTIFICACIONES

Yo, HENRY CRISTHIAN BELTRÁN AVEIGA, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Henry Cristhian Beltrán Aveiga

Certifico que el presente trabajo de integración curricular fue desarrollado por Henry Cristhian Beltrán Aveiga, bajo mi supervisión.

Igsil Augusto Dávila Montenegro

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Henry Cristhian Beltrán Aveiga

Igsil Augusto Dávila Montenegro

RESUMEN

La tarea de asignar justamente un conjunto de recursos, entre un grupo de agentes, es un problema de gran relevancia, y que, se ha visto un creciente interés en las áreas de salud, economía, finanzas, inteligencia artificial y sistemas multiagente.

Consideramos el problema de dividir equitativamente un conjunto de recursos. Gran parte de la literatura de división justa asume que los artículos son "bienes", es decir, producen una utilidad positiva para los agentes. También hay algunos trabajos en los que los elementos son "tareas" que generan una utilidad negativa para los agentes. En este trabajo, consideramos un escenario más general en el que un agente puede tener una utilidad negativa o positiva para cada artículo. Este marco captura, por ejemplo, la asignación justa de tareas, donde los agentes pueden tener utilidades tanto positivas como negativas para cada tarea. Mostraremos que mientras que algunos de los resultados axiomáticos y computacionales positivos se extienden a este entorno más general, otros no. Presentamos varios algoritmos nuevos y eficientes para encontrar asignaciones justas en este entorno general.

Palabras clave: Agentes, Recursos, Asignaciones, Repartición, Bienestar.

ABSTRACT

The task of fairly allocating a set of resources, among a group of agents, is a problem of great relevance, and one that has seen a growing interest in the areas of health, economics, finance, artificial intelligence and multi-agent systems.

We consider the problem of evenly dividing a set of resource. Much of the fair share literature assumes that items are "goods," that is, they produce positive utility for agents. There are also some jobs where items are "tasks" that generate negative utility for agents. In this paper, we consider a more general scenario in which an agent can have a positive or negative utility for each item. This framework captures, for example, fair task assignment, where agents can have both positive and negative utilities for each task. We will show that while some of the positive axiomatic and computational results extend to this more general setting, others do not. We present several new and efficient algorithms for finding fair allocations in this general environment.

Keywords: Agents, Resources, Assignments, Distribution, Welfare.

Índice general

Capítulo 1

Descripción del componente desarrollado

En el presente trabajo se propone estudiar el problema de asignación justa de bienes y tareas indivisibles. Si consideramos un grupo de trabajadores de una empresa a los cuales se les asigna diferentes tareas a realizar. Los trabajadores pueden tener puntos de vista subjetivos, con respecto a lo agradable que es realizar cada tarea, para algunas personas puede resultar en extremo fastidioso realizar alguna tarea fuera de la oficina, por lo que a este tipo de trabajador, pueden compensarse dándole algún premio valioso.

Así este problema puede verse como un problema de división donde los agentes tienen preferencia sobre los bienes o artículos y se requiere asignar los bienes o artículos de una manera justa. El giro que consideramos es que si un agente tiene utilidad positiva o negativa para un elemento es subjetivo.

Para realizar el presente trabajo se va a describir el problema general en el marco teórico y en la metodología se van a demostrar con detalle los teoremas y resultados importantes en el artículo [3].

1.1. Objetivo general

Desarrollar a detalle los resultados de la asignaciones justas, equitativas y libres de envidia cuando los bienes son una combinación de objetos subjetivos y tareas indivisibles.

1.2. Objetivos específicos

1. Proporcionar una base teórica detallada para el planteamiento del problema y explicación de las definiciones.

2. Estudiar el algoritmo de Double Round Robin que encuentra una asignación libre de envidia siempre que se tengan utilidades aditivas.

3. Reproducir de forma detallada los resultados principales del artículo [3].

1.3. Alcance

Entender y desarrollar en forma detallada las demostraciones de la teoría de asignación de recursos utilizando bienes y tareas indivisibles. Usando como referencias principales [10] y [3].

1.4. Marco teórico

En esta sección expondremos las definiciones y resultados generales que permiten comprender de manera precisa el problema de la repartición justa de recursos y que nos ayudaran al desarrollo de nuestro trabajo.

1.4.1. Agentes, recursos y asignaciones

Definición 1.4.1. *Los recursos o bienes son un conjunto finito O de m elementos $O = \{o_1, o_2, ..., o_m\}$.*

Los tipos de recursos de un problema de división justa, pueden ser de distintos índoles, dependiendo de la naturaleza del problema, por ejemplo;

Ejemplo 1.4.1. *Si deseamos repartir un postre a varios invitados en una fiesta. Los bienes se clasifican en dos tipos, bienes divisibles como barras de metal, sumas de dinero, alimentos, etc y los indivisibles como lo pueden ser un carro, la casa o cualquier bien inmueble, etc.*

Además de la distinción entre bienes divisibles e indivisibles, podemos clasificar los bienes de acuerdo a si son estáticos o si pueden potencialmente cambiar sus propiedades durante el proceso de asignación como por ejemplo;

Ejemplo 1.4.2. *La cantidad y cualidad de la mano de obra para la construcción de edificaciones.*

Definición 1.4.2. *Denotaremos al conjunto N como el conjunto de los agentes o entidades a las cuales se les asignara los recursos por lo que este conjunto también debe ser finito, es decir $|N| = n$, $N = \{1, 2, ..., n\}$.*

Ejemplo 1.4.3. *Los agentes pueden ser desde individuos, grupos de individuos, empresas, instituciones, etc.*

Definición 1.4.3. *Una asignación de m recursos a n agentes, puede verse como una función, es decir, cada recurso es entregado una única vez a algún agente.*

Definición 1.4.4. *Definición formal de Asignación Una asignación A de los recursos O en los agentes N o simplemente una asignación, es una partición de O, $\{A_1, A_2, ..., A_n\}$, representada a través de una n-úpla,*

$$A = (A_1, A_2, ..., A_n)$$

donde para cada $i \in N$, A_i representa los recursos asignados al agente i. El conjunto formado por todas las asignaciones se denota por N^O que gracias a los métodos y teoría de de conteo se puede probar que si $|N| = n$ y $|O| = m$ entonces todas las relaciones posibles entre los conjuntos es igual a n^m. Notemos que con la partición $\{A_1, A_2, ..., A_n\}$ de O, se garantiza que:

- *Todos los recursos son repartidos, es decir, $\forall o \in O$, $\exists i \in N$ tal que $o \in A_i$.*

- *Ningún recurso es repartido dos veces, esto significa que $A_i \cap A_j \neq \emptyset$*

La división justa trata de identificar una asignación deseable de bienes a agentes:para cada agente, necesitamos especificar qué elemento deben obtener o se lo puede pensar de manera abstracta en una asignación como un acuerdo. Modelamos las asignaciones como funciones que asignan agentes a paquetes de bienes. la naturaleza exacta del co-dominio depende de los detalles del modelo de división justa considerado.

Ejemplo 1.4.4. *sí una fabrica de autos desea vender las gamas de autos altas, medias y bajas a 2 casas comerciales de autos donde $N = \{1, 2\}$ por lo que se pueden hacer las siguientes asignaciones $A = \{A_1, A_2\} = \{\{altas, medias\}, bajas\}$ o $B = \{B_1, B_2\} = \{altas, \{medias, bajas\}\}$. Modelamos las asignaciones como funciones que asignan recursos o bienes a un grupo de agentes como por ejemplo si un grupo de talleres mecánicos de autos necesita repuestos la empresa encargada de dar esos repuestos los distribuirá de acuerdo a la necesidad de los clientes.*

Uno de los retos actuales en este campo es encontrar asignaciones que produzcan satisfacción social e individual entre los agentes. Este tipo de problema se conoce en la literatura como distribución justa de recursos.

Una manera de garantizar que las asignaciones cumplan un estándar de aceptación, por parte de los agentes, es pedirle información a los agentes en cuanto a sus preferencias, gustos y necesidades lo cual se mide con funciones reales llamadas funciones de utilidad.

Para evaluar las propiedades de eficiencia y justicia es necesario conocer las preferencias que tiene cada agente sobre los recursos, y esto puede hacerse usando un enfoque cuantitativo, en el que cada agente define una función de utilidad, describiendo la valuación que le asocia a cada subconjunto de recursos. Otro enfoque utilizado es el cualitativo, en el cual, cada agente define sus preferencias a través de preórdenes.

Definición 1.4.5. *Una función de utilidad es una función la cual mide la satisfacción o utilidad que recibe un agente. Sea u función de $\mathscr{P}(O)$ a los números reales, $u : \mathscr{P}(O) \longmapsto \mathbb{R}$ se conoce como función de utilidad y el número $u(R)$ representa la utilidad del conjunto R, además la función de utilidad cumple que:*

- *Es no negativa si, y solo si, $u(R) \geq 0$ para todo $R \in \mathscr{P}(O)$.*

- *Es positiva si, y solo si, u es no negativa y para todo $R \in \mathscr{P}(O)$ con $R \neq \emptyset$ se tiene que $u(R) \neq 0$.*

- *Es aditiva si, y solo si, u es no negativa, $u(\emptyset) = 0$, y para todo $R \in \mathscr{P}(O)$*

$$u(R) = \sum_{r \in R} u(\{r\})$$

- *Tiene presupuesto K, con $K \in \mathbb{R}$ y $K \geq 0$ si, y solo si, $u(O) = K$ y por tanto*

$$u(O) = \sum_{o \in O} u(\{o\}) = K$$

Observaciones:

- Cuando nos referimos a la función de utilidad de un agente en particular para medir su satisfacción usaremos la notación de $u_i(A_i)$.

- Se puede determinar gracias a teoría combinatoria que en un problema de asignación de m recursos a n agentes existen n^m posibilidades de asignaciones.

Ejemplo 1.4.5. *si deseamos repartir 2 bienes a 2 agentes, es decir, $O = \{a, b\}$ y $N = \{1, 2\}$, por lo que la función de de utilidad del agente 1 sera $u_1(\{a\}) = 1$ y $u_1(\{b\}) = 2$, claramente es aditiva puesto que es no negativa y además $u_1(\{\emptyset\}) = 0$ y tiene un presupuesto de 3, ya que, $u_1(O) = u_1(\{a\}) + u_1(\{b\}) = 3$, similarmente se puede hacer una función de utilidad para el agente 2.*

Ejemplo 1.4.6. *Consideremos los recursos $O = \{a, b, c, d, e\}$ y los agentes $N = \{1, 2, 3\}$, además veamos las siguientes asignaciones $A = (A_1, A_2, A_3) = (\{b, c, e\}, \{a, d\}, \{\emptyset\})$, por que tendríamos lo siguiente;*

$$u_1(A_1) = u_1(\{b, c, e\}) = \sum_{x \in A_1} u_1(x)$$

$$u_2(A_2) = u_1(\{a, d\}) = \sum_{x \in A_2} u_2(x)$$

$$u_3(A_3) = u_3(\{\emptyset\}) = 0$$

Lo cual nos quiere decir que el agente 3 en su asignación, no recibe ningún recurso, por lo que su función de utilidad por definición es cero.

Definición 1.4.6. *Diremos que dadas dos asignaciones $A, B \in \mathscr{P}(O)$ y un agente $i \in N$, la notación $A \succcurlyeq_i B$ significara que el agente i tiene mayor o igual preferencia de utilidad en la asignación A antes que la asignación B, es decir, $u_i(A) \geq u_i(B)$ y denotaremos $A \sim_i B$ cuando el agente tenga igual preferencia de utilidad en ambas asignaciones, es decir, $u_i(A) = u_i(B)$*

Como se evidenciara en los siguientes ejemplos.

Ejemplo 1.4.7. *En una cooperativa de taxis se ha realizado un torneo de cuarenta entre los miembros de la cooperativa, donde solo una pareja de competidores ha ganado los siguientes premios; \$50 en efectivo (t), un fin de semana libre (l), medallas de ser campeones (c) y derecho a una bebida alcohólica que ellos gusten (b). Para este caso tenemos que $N = \{1, 2\}$ y $O = \{t, l, c, b\}$, así sabemos que existen n^m posibles asignaciones y para nuestro caso $|N^O| = 16$. Supongamos que el ganador 1 considera la función de utilidad u_1, dada por*

$$u_1(\{t\}) = u_1(\{c\}) = u_1(\{b\}) = 3{,}5 \quad y \quad u_1(\{l\}) = 4{,}5$$

se tiene que

$$\{l\} \succ_1 \{t\} \sim_1 \{c\} \sim_1 \{b\}$$

Lo que quiere decir que el agente 1 tiene mayor preferencia por el recurso $\{l\}$ antes que el resto y las demás las considera iguales es decir no le importa cual de los recursos $\{t, c, b\}$ se le asigne.

Ahora el ganador 2 considera la función de utilidad aditiva dada por:

$$u_2(\{t\}) = u_2(\{l\}) = u_2(\{c\}) = u_2(\{b\}) = 3{,}75$$

Así,

$$\{t\} \sim_2 \{l\} \sim_2 \{c\} \sim_2 \{b\}$$

Podemos ver que tanto para u_1 y u_2 tiene presupuesto de $K = 15$. Ahora si tomamos la asignación $A = (\{t\}, \{l, c, b\})$, el ganador 2 evalúa lo que le han asignado con $3{,}75$; es decir, $u_2(A_1) = u_2(\{t\}) = 3{,}75$ y evalúa lo que le asignaron a el ganador 1; es decir, $u_2(A_2) = u_2(\{l, c, b\}) = 11{,}25$, por lo que el ganador 2 le da mas utilidad a lo que le asignaron al ganador 1 pues $u_2(A_2) > u_2(A_1)$.

Por otro lado, se asume que para todo $i \in N$ las preferencias individuales $\succcurlyeq_i$ son dadas a través de funciones de utilidad, u_i donde $\succcurlyeq_i$ esta dado por el lema a continuación.

Lema 1.4.1. *Dado un agente $i \in N$ y u_i una función de utilidad asociada a i , entonces la relación $\succcurlyeq_i$ sobre $\mathcal{P}(O)$ dada por : $\forall A, B \in \mathcal{P}(O)$,*

$$A \succcurlyeq_i B \Longleftrightarrow u_i(A) \geq u_i(B)$$

es una preferencia individual del agente i.

Demostración: Dado que se quiere probar que $\succcurlyeq_i$ es un preorden total. Sean $A, B, C \in P(O)$. Primero se probara la totalidad. Como $u_i(A), u_i(B) \in \mathbb{R}$, por la tricotomía de los números reales, se tiene que

$$u_i(A) \geq u_i(B)$$

O también,

$$u_i(B) \geq u_i(A)$$

luego por hipótesis $A \succcurlyeq_i B$ o $B \succcurlyeq_i A$. Así, $\succcurlyeq_i$ es total.

Para probar la transitividad supongamos que $A \succcurlyeq_i B$ y $B \succcurlyeq_i C$. Por hipótesis obtendremos que,

$$u_i(A) \geq u_i(B)$$

Y tambien que,

$$u_i(B) \geq u_i(C)$$

Por la transitividad de los números reales se sigue que $u_i(A) \geq u_i(C)$ y nuevamente por hipótesis obtenemos que $A \succcurlyeq_i C$. Asi $\succcurlyeq_i$ es transitiva por lo que se ha probado que $\succcurlyeq_i$ es total y transitiva y por tanto se concluye que $\succcurlyeq_i$ es un preorden total. ∎

Eficiencia y justicia

En este apartado se estudian propiedades para determinar cuándo una asignación es buena. Ya que estas propiedades están relacionadas con la eficiencia y la justicia.

La satisfacción social generalmente se mide a través de la propiedad de eficiencia de Pareto, también conocida como optimalidad de Pareto, la cual establece que la distribución de los recursos no puede ser cambiada para mejorar a un agente, sin perjudicar a otro.

Eficiencia - óptima de Pareto

Una asignación se denomina eficiente en el sentido de Pareto (u óptimo en el sentido de Pareto) si no existe ninguna otra asignación factible que mejoraría estrictamente la situación de al menos un agente sin empeorar la situación de ninguno de los demás. Esta satisfacción se mide comparando dos asignaciones A y B, a partir de las preferencias individuales de cada agente y sus adecuadas evaluaciones.

Definición 1.4.7. *Dados $A, B \in N^O$ y $i, j \in N$. X es Pareto dominada por Y si, $u_i(A_i) \leq u_i(B_i)$ y existe al menos un agente, j, donde $u_j(A_j) < u_j(B_j)$.*

Notemos que en la definición de Pareto dominada no requiere que podamos comparar utilidades entre diferentes agentes, ni hace ninguna referencia a las preferencias. De hecho, los órdenes de preferencia son simples.

Ejemplo 1.4.8. *Se tienen 3 camaroneras en distintas ciudades costeras del Ecuador en Santa Elena(S), Manta(M) y Esmeraldas(E), las cuales desean tener un control sobre las plagas que las afectan, para ello recuren a una empresa experta en control de plagas la cual proporciona 3 tipo de control, los cuales son, control de cianobacterias(T), control de cangrejos(C) y control de langostas negras(L). Así tenemos que los recursos son $O = \{T, C, L\}$ y los agentes $N = \{1, 2, 3\}$.*
Para cada agente $i \in N$ se tienen funciones de utilidad aditivas u_i dadas de la siguiente manera:

$$u_1(\{T\}) = 20;\ u_1(\{C\}) = 5;\ u_1(\{L\}) = 25$$

$$u_2(\{T\}) = 15;\ u_2(\{C\}) = 30;\ u_2(\{L\}) = 5$$

$$u_3(\{T\}) = 40; \; u_3(\{C\}) = 5; \; u_3(\{L\}) = 5$$

Como podemos ver todas las funciones de utilidad tiene presupuesto $K = 50$, *pues para cada agente* $i \in N$ *se tiene que* $u_i(O) = \sum_{o \in O} u_i(\{o\}) = 50$. *Considerando las siguientes asignaciones:*

$$A = (A_1, A_2, A_3) = (\{T, C\}, \{L\}, \emptyset)$$

$$B = (B_1, B_2, B_3) = (\{L\}, \{C\}, \{T\})$$

Así, las evaluaciones de cada camaronera de acuerdo a las asignaciones son las siguientes:

$$u_1(A_1) = u_1(\{T, C\}) = u_1(\{T\}) + u_1(\{C\}) = 25; \;\; u_1(B_1) = u_1(\{L\}) = 25$$

$$u_2(A_2) = u_2(\{L\}) = 5; \;\; u_2(B_2) = u_2(\{C\}) = 30$$

$$u_3(A_3) = u_E(\{\emptyset\}) = 0; \;\; u_3(B_3) = u_E(\{T\}) = 40$$

Podemos ver que en la ciudad de Santa Elena en las asignaciones A, B *tienen igual utilidad; pero, en* B *en las ciudades de Manta y Esmeraldas mejora su utilidad, por lo que* A *es pareto dominada por* B.

Definición 1.4.8 (Óptima de Pareto). *Dados* $A, B \in N^O$, *se dice que* B *es óptima de Pareto, PO, si no es Pareto dominada por otra asignación* A.*(Las asignaciones PO también son llamadas eficientes en el sentido de Pareto, o simplemente eficientes). Las asignaciones PO más fáciles de hallar son aquellas que reparten todos los recursos a un solo agente.*

Ejemplo 1.4.9. *Considerando el ejemplo anterior donde* $O = \{T, C, L\}$ *y* $N = \{1, 2, 3\}$, *sabemos que existen* $3^3 = 27$ *asignaciones posibles, por lo que si tomamos la asignación* $Z = \{O, \emptyset, \emptyset\}$. *En este caso,*

$$u_1(Z_1) = u_S(O) = 50; \;\; u_2(Z_2) = u_S(\emptyset) = 0; \;\; u_3(Z_3) = u_S(\emptyset) = 0$$

Puesto que la función de utilidad de la ciudad de Santa Elena es positiva, cualquier otra asignación que mejores a las funciones de las demás ciudades perjudica a la ciudad de Santa Elena; por lo que, la asignación Z *es* PO, *también puede encontrase otra asignación* PO *y comprobar que las 26 restantes no la dominan como podría ser el caso de la asignación* Y *en el* *ejemplo anterior.*

Justicia - Libre de envidia

Una asignación se dice que es libre de envidia si todos los agentes se valoran igual o mejor al conjunto de recursos asignados, que al conjunto de recursos asignados a los otros agentes. Esto describe la satisfacción de los agentes con la asignación. Desafortunadamente, en el caso de recursos indivisibles esta propiedad no siempre existe.

La satisfacción individual involucra criterios de justicia, que pueden evaluarse a través de la propiedad libre de envidia. Cuando esta propiedad se alcanza se dice que la asignación es libre de envidia (EF).

Definición 1.4.9. *Una asignación $A \in N^O$ es libre de envidia, EF, si para todo, $i, j \in N$, $u_i(A_i) \geq u_i(A_j)$, es decir, que el agente i considera la asignación A_i mejor que cualquier otra asignación otorgada a algún agente j. Obsérvese que si existe un agente $i \in N$ tal que $u_i(A_i) < u_j(A_j)$ para algun $j \in N$ entonces el agente i envidia al agente j. Por otro lado, se denota con EF al conjunto formado por todas las asignaciones EF.*

Ejemplo 1.4.10. *Supongamos que el área de salud de la ciudad de Quito divididas en entre publicas(1) y privadas (2) desean contratar equipos médicos a una empresa que proporciona 3 combos de equipos médicos (E_1, E_2, E_3), donde claramente tenemos que $O = \{E_1, E_2, E_3\}$ y $N = \{1, 2\}$, con funciones de utilidad aditivas dadas por:*

$$u_1(\{E_1\}) = 10 \text{ , } u_1(\{E_2\}) = 20 \text{ y } u_1(\{E_3\}) = 40$$

$$u_2(\{E_1\}) = 14 \text{ , } u_2(\{E_2\}) = 40 \text{ y } u_2(\{E_3\}) = 26$$

Con las siguientes asignaciones,

$$A = (A_1, A_2) = (\{E_1, E_3\}, \{E_2\})$$

$$B = (B_1, B_2) = (\{E_2\}, \{E_1, E_3\})$$

Ahora evaluando las asignaciones para cada agente se tiene que,

$$u_1(A_1) = u_1(\{E_1, E_3\}) = u_1(\{E_1\}) + u_1(\{E_3\}) = 10 + 40 = 50 \text{ y}$$
$$u_1(A_2) = u_1(\{E_2\}) = 20$$

$$u_2(A_1) = u_2(\{E_1, E_3\}) = u_2(\{E_1\}) + u_2(\{E_3\}) = 14 + 16 = 40 \text{ y}$$
$$u_2(A_2) = u_2(\{E_2\}) = 40$$

Luego,

$$50 = u_1(A_1) > u_1(A_2) = 20$$

$$40 = u_2(A_1) > u_2(A_2) = 40$$

En este caso ningún agente envidia al otro por lo que $A \in EF$, mientras que para la asignación B se tiene que,

$$u_1(B_1) = 20, \; u_1(B_2) = 50$$

$$u_2(B_1) = 40, \; u_2(B_2) = 40$$

Por lo que;

$$u_1(B_1) < u_1(B_2) \text{ y } u_2(B_1) = u_2(B_1)$$

Asi el agente 1 envidia al agente 2 por lo que $B \notin EF$.

Puesto que resulta bastante complicado satisfacer a todos lo agentes sin que ninguno envidie a otro se propone relajar este concepto de libre de envidia a uno nuevo que se define a continuación.

Libre de envidia de hasta un bien (EF1)

La propiedad libre de envidia de hasta un recurso es una propiedad que establece cuando la envidia entre los agentes es mínima; en el sentido que, si existe envidia entre un par de agentes, la misma desaparece eliminando un recurso. Vale aclarar que, cuando se dice eliminar un recurso, no se hace referencia a quitar ese recurso del problema; esto se refiere a que se garantiza que los agentes están a un recurso de estar satisfecho con los recursos asignados.

Definición 1.4.10. *Una asignación $A \in N^O$ se dice que es libre de envidia de hasta un bien, EF1, si para todo par de agentes $i, j \in N$, existe $g \in A_j$ tal que $u_i(A_i) \geq u_i(A_j - \{g\})$.*

Se denota por $\mathcal{EFO}$ al conjunto formado por todas las asignaciones $EF1$.

Ejemplo 1.4.11. *Del ejemplo anterior pudimos ver que $B \notin EF$, ahora veamos que la asignación B puede estar en el conjunto $EF1$. En efecto, el agente 1 envidia al agente 2.*

$$u_1(B_1) < u_1(B_2)$$

Se debe determinar que recurso en B_2 se puede suprimir para eliminar la envidia. Si quitamos el recurso $\{E1\}$, vemos que $u_1(B_1) = 20$ y $u_1(B_2/\{E1\}) = u_1(\{E3\}) = 40$ la envidia no desaparece, mientras que si quitamos el recurso $\{E3\}$, entonces la envidia desaparece pues nos quedaría; $u_1(B_1) = 20$ y $u_1(B_2/\{E3\}) = u_1(\{E1\}) = 10$. Luego, el recurso que genera envidia en la asignación B es $\{E3\}$, por lo que el conjunto $B/\{E3\} \in EF1$.

Hallar asignaciones que satisfacen las propiedades descritas es un problema difícil, debido a que el conjunto de asignaciones aumenta a medida que crece el número de agentes o de recursos. Por esta razón, se clasifican las asignaciones a través de una función denominada bienestar social donde estas tratan de capturar la satisfacción grupal. Si se desea realizar un análisis de los criterios de justicia y eficiencia comparando el bienestar social de cada par de asignaciones, es complicado y en la mayoría de casos innecesario puesto que el conjunto de todas las posibles asignaciones puede ser muy grande. Por este motivo, se desea reducir el tamaño del espacio de búsqueda y tomar en cuenta el conjunto formado por todas las asignaciones que maximizan a algunas funciones de bienestar.

En principio, podemos tener en cuenta todo tipo de indicadores al juzgar la equidad. Una postura particular a adoptar sería decir que la única información que deberíamos usar son los niveles de utilidad de los agentes individuales. Esto se conoce como el enfoque de bienestar. Técnicamente, esto significa que, en lugar de mirar las asignaciones y evaluar su equidad relativa.

Funciones de bienestar social

Un bienestar social natural es el bienestar social unitario, el cual se define como la suma de las utilidades que los agentes dan a lo asignado. Por lo que las funciones de bienestar social, son métodos para ordenar las asignaciones usando las preferencias individuales de los agentes. Estos métodos consisten en definir relaciones binarias $\succcurlyeq$ sobre N^O, Para ello se definen las funciones de bienestar social.

Definición 1.4.11. *Una función de bienestar sobre N^O es cualquier función $SW : N^O \longrightarrow \mathbb{R}$, por sus siglas en ingles(social welfare), donde el número $SW(A)$ representa el bienestar social de la asignación A. Estas funciones permiten ordenar a todas las posibles asignaciones de tal manera que se puede etiquetar cual es la mejor asignación entre todas las demás.*
Dada una SW se define la relación bienestar social $\succcurlyeq$ de la siguiente manera: para todo par $A, B \in N^O$,

$$A \succcurlyeq B \Longleftrightarrow SW(A) \geq SW(B)$$

donde $A \succcurlyeq B$ es interpretado como "la asignación A es socialmente más preferida o igualmente preferida que "la asignación B".

Toda la proposición interpreta que, la asignación A es socialmente más preferida o igualmente preferida que la asignación B si y solo si la asignación A tiene mayor o igual bienestar social que la asignación B.

Lema 1.4.2. *Dada SW una función de bienestar, entonces la relación $\succcurlyeq$ sobre $\mathcal{P}(O)$ dada por : $\forall A, B \in \mathcal{P}(O)$,*

$$A \succcurlyeq B \Longleftrightarrow SW(A) \geq SW(B)$$

es una preferencia grupal.

Demostración: Dado que se quiere probar que $\succcurlyeq$ es un preorden total. Sean $A, B, C \in P(O)$. Primero se probara la totalidad. Como $SW(A), SW(B) \in \mathbb{R}$, por la tricotomía de los números reales, se tiene que

$$SW(A) \geq SW(B)$$

O también,

$$SW(B) \geq SW(A)$$

luego por hipótesis $A \succcurlyeq B$ o $B \succcurlyeq A$. Así, $\succcurlyeq$ es total.

Para probar la transitividad supongamos que $A \succcurlyeq B$ y $B \succcurlyeq C$. Por hipótesis obtendremos que,

$$SW(A) \geq SW(B)$$

Y tambien que,

$$SW(B) \geq SW(C)$$

Por la transitividad de los números reales se sigue que $SW(A) \geq SW(C)$ y nuevamente por hipótesis obtenemos que $A \succcurlyeq C$. Asi $\succcurlyeq$ es transitiva por lo que se ha probado que $\succcurlyeq$ es total y transitiva y por tanto se concluye que $\succcurlyeq$ es un preorden total. ■

Una bondad que tiene el bienestar social utilitario es que toda asignación que maximice este bienestar social es PO ; lo malo es que, no siempre son $EF1$.

A continuación enunciamos algunas funciones de bienestar social clásicas en la teoría de repartición justa de recursos:

Bienestar social utilitario

Definición 1.4.12. *Para todo $A \in N^O$, el bienestar social utilitario, SW_U de A está definido como:*

$$SW_U(A) = \sum_{i \in N} u_i(A_i)$$

donde para todo $i \in N$, u_i es la función de utilidad.
Se denota por $\succcurlyeq_U$ la relación de bienestar social que define a partir de SW_U según la ecuación anterior de funciones de bienestar social.

Esto nos permite ordenar todas las asignaciones posibles de un problema, basándonos en el bienestar utilitario de cada asignación.

Ejemplo 1.4.12. *Supongamos que una pastelería debe repartir 3 tipos de pasteles de chocolate (Ch), vainilla (V) y naranja (Na) a una fiesta de bodas 1 y a una fiesta de grado 2, es decir, $O = \{Ch, V, Na\}$ y $N = \{1, 2\}$,* *con funciones de utilidad aditivas dadas como:*

$$u_1(\{Ch\}) = 20;\ u_1(\{V\}) = 30;\ u_1(\{Na\}) = 35$$

$$u_2(\{Ch\}) = 25;\ u_2(\{V\}) = 30;\ u_2(\{Na\}) = 30$$

Con las siguientes asignaciones,

$$A = (\{V\}, \{Ch, Na\});\ B = (\{Na\}, \{Ch, V\});\ C = (O, \emptyset);\ D = (\{Ch\}, \{V, Na\})$$

Luego,

$$SW_U(A) = u_1(\{V\}) + u_2(\{Ch, Na\}) = 30 + 55 = 85$$
$$SW_U(B) = u_1(\{Na\}) + u_2(\{Ch, V\}) = 30 + 60 = 90$$
$$SW_U(C) = u_1(\{Ch, V, Na\}) + u_2(\{\emptyset\}) = 85 + 0 = 85$$
$$SW_U(D) = u_1(\{Ch\}) + u_2(\{V, Na\}) = 20 + 60 = 80$$

Así,

$$SW_U(B) > SW_U(A) = SW_U(C) > SW_U(D)$$

Por lo que se tiene,

$$B \succ_U A \succ_U C \succ_U D$$

Se concluye que la asignación B produce más bienestar social utilitario que A, C y D, mientras que D produce el menos bienestar social de las asignaciones.

observación: *Según el bienestar social utilitario la asignación B es la mejor asignación entre A, C y D.*

Un acuerdo con el máximo bienestar social utilitario es un acuerdo que maximiza la utilidad media, lo que explica por qué este puede considerarse un criterio social atractivo. Por otro lado, esta definición de bienestar social ignora por completo las consideraciones de equidad:

Ejemplo 1.4.13. *Una asignación que dé una utilidad de 101 a un agente y 0 a otro se consideraría socialmente superior a una asignación que dé a ambos una utilidad de 50.*

Bienestar social de Nash

Al contrario de las asignaciones que maximizan el bienestar social utilitario, toda asignación que maximice el bienestar social de Nash es eficiente y justa.

Definición 1.4.13. *Sea $\forall A \in N^O$ el bienestar social de Nash de A es definida como:*

$$SW_{Nash}(A) = \prod_{i \in N} u_i(A_i)$$

donde para todo $i \in N$, u_i es una función de utilidad.

La relación de bienestar social que se define, según la **Definición 1.4.7**, a partir de SW_{Nash} se denota por $\succcurlyeq_{Nash}$.

Las asignaciones que maximizan el bienestar social utilitario son óptimas de Pareto; sin embargo, generalmente no son justas, ya que puede ocurrir que solo algunos agentes obtengan todos los recursos y el resto ningún recurso. Otro índice para el bienestar social(SW) es el bienestar social de Nash, este se obtiene multiplicando las utilidades de los recursos asignados a cada agente. En el cual se demuestra que las asignaciones que maximizan el SW de Nash son óptimas de Pareto y débilmente justas, en el sentido que cumplen la propiedad de libre de envidia de hasta un recurso.

Ejemplo 1.4.14. *Consideremos la información del* **Ejemplo 1.4.12** *y veamos el bienestar de Nash.*

$$SW_{Nash}(A) = u_1(\{V\}) * u_2(\{Ch, Na\}) = 30 * 55 = 1650$$

$$SW_{Nash}(B) = u_1(\{Na\}) * u_2(\{Ch, V\}) = 30 * 60 = 1800$$

$$SW_{Nash}(C) = u_1(\{Ch, V, Na\}) * u_2(\{\emptyset\}) = 85 * 0 = 0$$

$$SW_{Nash}(D) = u_1(\{Ch\}) * u_2(\{V, Na\}) = 20 * 60 = 1200$$

Así,

$$SW_{Nash}(B) > SW_{Nash}(A) > SW_{Nash}(D) > SW_{Nash}(C)$$

Según el bienestar social de Nash la asignación B es la mejor asignación entre A, B, C y D.

Bienestar social igualitaria

Definición 1.4.14. *Para todo $A \in N^O$, el bienestar social utilitario, SW_{Egal} por sus siglas en ingles(Egalitarian) de A está definido como:*

$$SW_{Egal}(u) = \text{mín}\{u_i | i \in N\}$$

Es decir, maximizar el bienestar social igualitario equivale a aumentar la utilidad del miembro más desfavorecido de la sociedad (quienquiera que termine siendo) tanto como sea posible. Si tomamos como indicador la utilidad máxima en lugar de la mínima, obtenemos una forma elitista de bienestar social.

Ejemplo 1.4.15. *Consideremos de nuevo la información del* **Ejemplo 1.4.12** *y veamos el bienestar social igualitario.*

$$SW_{Egal}(A) = \text{mín}\{u_1(\{V\}), u_2(\{Ch, Na\})\} = \text{mín}\{30, 55\} = 30$$

$$SW_{Egal}(B) = \text{mín}\{u_1(\{Na\}), u_2(\{Ch, V\})\} = \text{mín}\{30, 60\} = 30$$

$$SW_{Egal}(C) = \text{mín}\{u_1(\{Ch, V, Na\}), u_2(\{\emptyset\})\} = \text{mín}\{85, 0\} = 0$$

$$SW_{Egal}(D) = \text{mín}\{u_1(\{Ch\}), u_2(\{V, Na\})\} = \text{mín}\{20, 60\} = 20$$

Así,

$$SW_{Egal}(A) = SW_{Egal}(B) > SW_{Egal}(D) > SW_{Egal}(C)$$

Según el bienestar social igualitario A, B son las mejores asignaciones entre A, B, C y D.

Bienestar social Elitista

Definición 1.4.15. *Para todo $A \in N^O$, el bienestar social utilitario, SW_{Elit} por sus siglas en ingles(Elitist) de A está definido como:*

$$SW_{Elit}(u) = \text{máx}\{u_i | i \in N\}$$

Claramente, el bienestar social elitista tiene poco en común con cualquier noción intuitiva de equidad. En una asignación en la que lo único que importa es que al menos un agente logre su objetivo, puede ser sin embargo la perfecta formalización del deseo social de un estado.

Ejemplo 1.4.16. *Consideremos de nuevo la información del* ***Ejemplo 1.4.12*** *y veamos el bienestar social igualitario.*

$$SW_{Elit}(A) = \text{máx}\{u_1(\{V\}), u_2(\{Ch, Na\})\} = \text{máx}\{30, 55\} = 55$$

$$SW_{Elit}(B) = \text{máx}\{u_1(\{Na\}), u_2(\{Ch, V\})\} = \text{máx}\{30, 60\} = 60$$

$$SW_{Elit}(C) = \text{máx}\{u_1(\{Ch, V, Na\}), u_2(\{\emptyset\})\} = \text{máx}\{85, 0\} = 85$$

$$SW_{Elit}(D) = \text{máx}\{u_1(\{Ch\}), u_2(\{V, Na\})\} = \text{máx}\{20, 60\} = 60$$

Así,

$$SW_{Elit}(C) > SW_{Elit}(B) = SW_{Elit}(D) > SW_{Elit}(A)$$

Según el bienestar social elitista la asignación C es la mejor asignación *entre A, B, C y D.*

Capítulo 2

Metodología

El método que usaremos en este trabajo es el inductivo-deductivo. Ahora nuestro modelo lo definimos como un problema de división justa de elementos indivisibles donde los agentes pueden tener utilidades tanto positivas como negativas. Donde a la tripleta $I = (N, O, U)$ la llamaremos instancia donde N es el conjunto de agentes, O conjunto de bienes y U es la n-tupla de funciones de utilidad $u_i : O \longmapsto \mathbb{R}$.

En nuestro modelo, un recurso puede ser bueno para un agente es decir $(u_i(o) > 0)$ pero una tarea para otro agente puede ser $(u_i(o) < 0)$, donde $o \in O$. Para $X \subseteq O$, escribiremos $u_i(X) = \sum_{o \in X} u_i(o)$; asumiremos que las funciones de utilidad son aditivas a no ser que se especifique lo contrario. A cada $X \subseteq O$, nos referiremos como un paquete de recursos y las asignaciones A_i con $i \in N$ serán definidas como anteriormente se mencionaron.

Primero observamos que todas las definiciones pueden extenderse a este modelo general. Otra noción atractiva de equidad es la proporcionalidad que garantiza a cada agente una fracción $1/n$ de su utilidad para todos los recursos.

Definición 2.0.1 (Proporcionalidad(PROP)). *Una asignación A es proporcional si cada agente $i \in N$ recibe una paquete A_i de valor al menos de su parte justa proporcional $u_i(O)/n$.*

Ejemplo 2.0.1. *Consideremos $N = \{1, 2\}$ y $O = \{a, b, c\}$ con las funciones* *de utilidad dadas por;*

$$u_1(\{a\}) = 10;\ u_1(\{b\}) = 5;\ u_1(\{c\}) = 5$$

$$u_2(\{a\}) = 6;\ u_2(\{b\}) = 12;\ u_2(\{c\}) = 2$$

Como vemos ambos agentes tienen un presupuesto de 20 *cada uno por lo que* $u_1(O) = u_2(O) = 20$*, ahora consideremos la asignación* $A = (A_1, A_2) = (\{a\}, \{b, c\})$*, veamos si es proporcional;*

$$u_1(A_1) = u_1(\{a\}) \geq \tfrac{u_1(O)}{n}$$

Pues $10 \geq \frac{20}{2}$*, mientras que para el agente* 2 *se tiene;*

$$u_2(A_2) = u_2(\{b, c\}) \geq \tfrac{u_2(O)}{n}$$

Así, $14 > \frac{20}{2}$*, por lo que se concluye que la asignación* A *es proporcional.*

De donde gracias a la definición de proporcionalidad en las asignaciones se tiene la siguiente proposición.

Proposición 2.0.1. *Una asignación completa libre de envidia satisface la proporcionalidad.*

Demostración: Supongamos que la asignación A es una asignación libre de envidia, por lo que si consideramos un agente $i \in N$, entonces por la definición de libre-envidia(EF) se tiene que;

$$u_i(A_i) \geq u_i(A_j), \forall j \in N$$

Entonces, sumando a ambos lados de la inecuación tenemos que;

$$\sum_{j \in N} u_i(A_i) \geq \sum_{j \in N} u_i(A_j) = u_i(O)$$

$$n * u_i(A_i) \geq \sum_{j \in N} u_i(A_j) = u_i(O)$$

Por lo tanto cada $i \in N$ recibe un paquete de utilidad de al menos $u_i(O)/n$ y en consecuencia la asignación A es proporcional. ■

Un ejemplo simple en el cual es bastante complicado lograr la ausencia de envidia y proporcionalidad es si solo existen dos agentes a los que se deben repartir los bienes.

Ahora presentamos una definición generalizada para (EF1) considerando que la envidia puede disminuir eliminando unos de los "bienes"del paquete de otro o una "tarea"del propio paquete.

Definición 2.0.2 (EF1). *Una asignación A es libre de envida de hasta un bien (EF1) si para todo $i, j \in N$, donde i no envidia a j o hay un recurso $o \in A_i \cup A_j$ tal que $u_i(A_i - \{o\}) \geq u_i(A_j - \{o\})$.*

Conitzer et al. [2017] introdujo una nueva relajación de la proporcionalidad, a la que llamaron PROP1. En el contexto de una buena asignación, esta relajación de la equidad es un debilitamiento tanto de EF1 como de proporcionalidad, requiriendo que cada agente obtenga su parte proporcional justa si obtiene un bien adicional de los demás. Ahora extenderemos esta definición a nuestro entorno: bajo nuestra definición, cada agente recibe su parte proporcional justa mediante la obtención de un bien adicional o la eliminación de alguna tarea de su paquete.

Definición 2.0.3 (PROP1). *Una asignación A satisface la proporcionalidad de hasta un bien(PROP1) si para cada $i \in N$,*

- $u_i(A_i) \geq u_i(O)/n$; *o*
- $u_i(A_i) + u_i(o) \geq u_i(O)/n$ *para algún* $o \in O - A_i$; *o*
- $u_i(A_i) - u_i(o) \geq u_i(O)/n$ *para algún* $o \in A_i$.

Proposición 2.0.2. *Una asignación $EF1$ completa satisface la PROP1.*

Demostración: Primero supongamos que $|N| \geq 2$ y $O \neq \emptyset$ y A una asignación libre de envidia de hasta un bien, para algún agente $i \in N$;

1. Considerando que $A_i = O$.
   - si $u_i(O) \geq 0$ entonces $u_i(A_i) = u_i(O) \geq u_i(O)/n$.

- si $u_i(O) < 0$, consideremos un agente $j \in N - \{i\} \neq \emptyset$, puesto que A es una asignación libre de envia de hasta un bien se sigue que hay un recurso $o* \in A_i$ tal que;

$$u_i(A_i) - u_i(o*) \geq u_i(A_j) = u_i(\emptyset) = o$$

$$u_i(A_i) - u_i(o*) \geq 0 > u_i(O)/n$$

2. Ahora considerando que $A_i = \emptyset$.

- Si $u_i(O) \leq 0$ entonces se sigue directamente que $u_i(A_i) = u_i(\emptyset) = 0 \geq u_i(O)/n$.
- Supongamos que $u_i(O) > 0$, así puesto que A es una asignación libre de hasta un bien entonces $\forall j \in N - \{i\}$ se tiene que hay un $o* \in A_j$ tal que;

$$u_i(A_i) \geq u_i(A_j - \{*o\})$$

lo que implica que,

$$u_i(A_i) = u_i(\emptyset) = 0 \geq u_i(A_j) - u_i(*o)$$

Sea $\bar{o} \in O$ tal que $\bar{o} \in \mathit{argmax}_{o \in O} u_i(o)$. Notar que $u_i(\bar{o}) > 0$ ya que $u_i(O) > 0$. Entonces;

$$u_i(A_i) - u_i(\bar{o}) \geq u_i(A_j) \quad \forall j \in N$$

lo que implica que,

$$u_i(A_i) - u_i(\bar{o}) \geq u_i(O)/n$$

Por ultimo, consideremos el caso cuando $O/A_i \neq \emptyset$ y $A_i \neq \emptyset$. Sea $x = \max_{o \in O/A_i} u_i(o)$ y $y = \min_{o \in A_i} u_i(o)$, Como se cumple EF1, para cualquier agente $j \in N - \{i\}$, se tiene que,

- i no envidia a j;
- Existe un elemento $o \in A_j$ tal que $u_i(A_i) \geq u_i(A_j) - u_i(o)$;
- Existe un elemento $o \in A_i$ tal que $u_i(A_i) - u_i(o) \geq u_i(A_j)$.

Lo cual implica lo siguiente:

- $u_i(A_i) \geq u_i(A_j)$;

- $u_i(A_i) + x \geq u_i(A_j)$;
- $u_i(A_i) - y \geq u_i(A_j)$.

Por lo tanto, si i obtiene una utilidad adicional $b^* := \text{máx}\{x, -y, 0\}$ al obtener algún recurso o eliminar alguna tarea, su utilidad actualizada es tal que $u_i(A_i) + b^* \geq u_i(A_j)$ para cualquier $j \in N - \{i\}$. Lo cual implica lo siguiente;

$$n * (u_i(A_i) + b^*) \geq \sum_{j \in N} u_i(A_j) = u_i(O)$$

Por lo tanto $u_i(A_i) + b^* \geq \frac{u_i(O)}{n}$ cumple con la PROP1. ∎

Ejemplo 2.0.2. *Consideremos el siguiente conjunto de agentes $N = \{1, 2, 3, 4\}$ y recursos $O = \{o_1, o_2, ..., o_9\}$. Las funciones de utilidad de los agentes están representadas en la siguiente tabla.*

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9
Agente 1:	*1*	*-1*	*2*	*1*	*-2*	*-4*	*-6*	*-1*	*-1*
Agente 2:	*4*	*-3*	*6*	*2*	*-2*	*-2*	*-2*	*-1*	*-1*
Agente 3:	*0*	*11*	*8*	*11*	*0*	*0*	*0*	*10*	*0*
Agente 4:	*0*	*11*	*8*	*11*	*0*	*0*	*0*	*0*	*10*

Tabla de funciones de utilidad de recursos y tareas subjetivos.

También consideramos la siguiente asignación A en la que;

$$A_1 = \{o_2, o_4\}$$

$$A_2 = \{o_1, o_3, o_5, o_6, o_7\}$$

$$A_3 = \{o_8\}$$

$$A_4 = \{o_9\}$$

Donde la utilidad resultante de los agentes es la siguiente;

$$u_1(A_1) = u_1(o_2) + u_1(o_4) = 0,$$

$$u_2(A_2) = u_2(o_1) + u_2(o_3) + u_2(o_5) + u_2(o_6) + u_2(o_7) = 4,$$

$$u_3(A_3) = u_3(o_8) = 10,$$

$$u_4(A_4) = u_4(o_9) = 10$$

Como podemos ver la asignación A satisface la proporcionalidad ya que $u_i(A_i) \geq \frac{u_i(O)}{n}$, para todo $i \in N$, sin embargo, A no satisface la ausencia de envidia ya que los agentes 3 y 4 tienen envidia del agente 1 puesto que $u_3(A_3) < u_3(A_1)$ y $u_4(A_4) z u_4(A_1)$. La asignación A incluso viola la propiedad de EF1 ya que los agentes 3 y 4 envidian al agente 1 por más de un elemento.

La asignación no es óptima en el sentido de Pareto ya que los agentes 1 y 2 obtienen elementos o5, o6, o7, para los cuales tienen una utilidad negativa. Estos artículos se pueden dar al agente 3 o al 4, ya que no tiene ninguna utilidad para ellos.

2.1. Encontrando una asignación libre de envidia de hasta un bien

En esta sección nos centraremos en las asignaciones libre de envidia de hasta un bien el cual es un concepto muy permisivo que admite un algoritmo de tiempo polinomial en el caso de una buena asignación. Se presentaran dos algoritmos que generalizan apropiadamente el concepto de libre de envidia de hasta un bien, la regla de Round Robin y el algoritmo de Envy Graph [14].

2.1.1. Algoritmo Doble Round Robin

Se considera una regla de turno rotativo en la que los agentes se turnan y eligen su elemento no asignado preferido. La regla de turno rotativo encuentra una asignación libre de envidia de hasta un bien, si todos los artículos son bienes y siguiendo esta misma lógica también encuentra una asignación libre de envidia de hasta un bien si se trata solamente de tareas. Sin embargo, se presentara una proposición que muestra que la regla no se cumple si existen bienes y tareas a la vez, pues ya no encuentra una asignación EF1.

Ejemplo 2.1.1. *Consideremos el caso donde $N = \{1, 2\}$ y $O = \{o_1, o_2, o_3\}$ y en la siguiente tabla las funciones de utilidad respectivas;*

	o_1	o_2	o_3
Agente 1:	*1*	*3*	*5*
Agente 2:	*3*	*1*	*4*

Además si la asignación esta dada por $A = (A_1, A_2) = (\{o_2, o_3\}, \{o_1\})$, *luego vemos que,* $u_2(o_3) < u_1(o_3)$.

Siguiendo la regla de turno rotativo, si el agente 1 *es el primero en elegir, como vemos elegiría el que tiene mayor utilidad, es decir, el recurso* o_3 *y el agente* 2 *tendría que elegir otro recurso de su preferencia, pero aun así seguiría teniendo envidia del agente* 1, *mientras que si ahora el agente* 2 *fuese el primero en elegir. Este elegiría el recurso* o_3, *por tanto, el agente* 1 *se vería obligado a elegir otro recurso y como vemos en la tabla, de ser este el caso el agente* 1 *sentiría envidia del agente* 2.

Así, si eliminamos el recurso o_3 *tendríamos una nueva asignación* $A - \{o_3\} = (\{o_2\}, \{o_1\})$, *de donde es claro que la envidia desaparecería para ambos agentes.*

Proposición 2.1.1. *La regla de Round Robin no satisface la libre envidia de hasta un bien.*

Demostración: Supongamos que hay dos agentes y cuatro artículos con utilidades idénticas que se describen en la Tabla 1. Consideremos el orden en el que el agente M elige el único bien y luego las tareas restantes de igual valor se asignan en consecuencia. En ese caso, el agente M obtiene un bien de alto valor y una tarea, mientras que el agente R recibe dos tareas. Entonces, incluso si se elimina un artículo de los paquetes de M o R, R seguirá teniendo envidia.

	(1)	(2)	(3)	(4)
M, R:	5	-8	-8	-8

Tabla 1

■

Una cuidadosa adaptación del método round-robin a nuestro entorno, al que llamamos algoritmo double round-robin, construye una asignación libre de envidia de hasta un bien. En esencia, el algoritmo aplicará el método de round-robin dos veces: en el sentido de las agujas del reloj y en el sentido contrario. En la primera fase, el algoritmo round-robin asigna los elementos para los que cada agente tiene una utilidad no positiva, mientras que en la segunda fase, el algoritmo round-robin inverso asigna a los agentes los elementos restantes para los que algún agente tiene una utilidad positiva, en el orden inverso comenzando por el agente que elige último en la primera fase. Intuitivamente, cada agente i puede envidiar al agente j que llega antes que él al final de una fase, pero i no envidia a j con respecto a los elementos asignados en la otra ronda; por lo tanto, la envidia de i hacia j se puede acotar a un elemento. Se muestra a continuación el algoritmo.

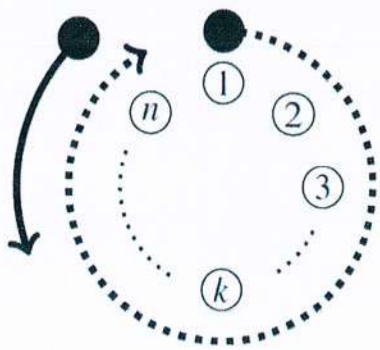

Figura 1: Regla Double Round-Robin.

En la **Figura 1** del algoritmo Double Round Robin. La línea punteada corresponde al orden de selección al asignar las tareas. La línea gruesa corresponde al orden de preparación al asignar recursos. El círculo negro sólido indica el agente que comienza con el picking. Para la ronda punteada (tareas), el agente 1 es el primer agente en elegir. Para la ronda sólida (recursos), el agente n es el primer agente en elegir.

Algorithm 1 Algoritmo Doble Round Robin

Input: Instancia $I = (N, O, U)$

Output: Una asignación A

1: Dividir O en $O^+ = \{o \in O | \exists i \in N : u_i(o) > 0\}$, $O^- = \{o \in O | \exists i \in N : u_i(o) \leq 0\}$. Suponer que $|O^-| = an - k$ para algun entero $a, k \in \{0, ..., n-1\}$.

2: Crear k tareas ficticias para las que cada agente tiene utilidad 0 y agregarlos al conjunto O^-.(Por lo tanto $|O^-| = an$).

3: Dejar que los agentes entren en una secuencia de Round Robin $(1, ..., n)^*$ y elijan su elemento preferido en O^- hasta que se asignen todos los elementos en O^-.

4: Dejar que los agentes entren en una secuencia de Round Robin $(n, n-1, ..., 1)^*$ y elijan su elemento preferido en O^+ hasta que se asignen todos los elementos en O^+. Si un agente no tiene un artículo disponible que se le dé una utilidad estrictamente positiva, no obtiene un artículo real, sino que pretende elegir uno ficticio para el que tiene una utilidad de 0.

5: Eliminar los elementos ficticios de la asignación actual A y devolver la asignación resultante A^*.

A continuación, para una asignación A y una asignacion X, decimos que i envidia a j con respecto a X si $u_i(A_i \cap X) < u_i(A_j \cap X)$.

Teorema 2.1.1. *El algoritmo double round robin (Algoritmo 1) devuelve una asignación completa de libre de envidia de hasta un bien en tiempo* $O(\text{máx}\{m^2, mn\})$.

Demostración: Notemos que el algoritmo asegura que todos los agentes reciben el mismo número de tareas si se introducen k tareas.

Sea A la asignación resultante del algoritmo 1. Veamos que A satisface la libre envidia de hasta un bien, para ello consideremos un par de agentes $i, j \in N$ donde $i < j$. Mostraremos que al eliminar un elemento del paquete de i o del paquete de j, estos agentes no se envidiarán entre sí. Denotamos x_k^i y x_k^j como los k-ésimos elementos asignados al agente i y al agente j para $k = 1, 2, ..., a$ en la Línea 4, respectivamente. Denotamos y_k^i y y_k^j como los k-ésimos elementos asignados al agente i y al agente j para $k = 1, 2, ..., b$ en la Línea 5, respectivamente, donde b denota el número de rondas en las que cada agente elige un elemento (incluyendo un elemento ficticio) en la línea 5.

1. Consideremos que i envidia a j donde vemos que al k-recurso $x_k^i \in$

O^- asignado a i es poco preferido por i al k-recurso $x_k^j \in O^-$ asignado a j, por lo que i no envidia a j con respecto a O^- es decir,

$$u_i(A_i \cap O^-) = \sum_{k=1}^{a} u_i(x_k^i) \geq \sum_{k=1}^{a} u_i(x_k^j) = u_i(A_j \cap O^-) \tag{2.1}$$

Mientras que para la asignación de recursos de O^+, el agente i puede envidiar al agente j en O^+, pero si el primer recurso elegido y_1^j de j en O^+ se elimina de la asignación para j entonces la envidia desaparecerá es decir i no envidiara a j con respecto a $O^+ - \{y_1^j\}$ de donde;

$$u_i(A_i \cap O^+) = \sum_{k=1}^{b} u_i(y_k^i) \geq \sum_{k=1}^{b} u_i(y_k^j) = u_i((A_j \cap O^+) - \{y_1^j\}) \tag{2.2}$$

Esto se debe a que cada recurso y_k^j elegido por j donde $k = 2, 3, ..., b$, hay un elemento correspondiente y_{k-1}^i elegido por i antes del turno de j que es poco preferido por i para y_k^i, por lo que de (2.1) y (2.2) se sigue que $u_i(A_i) \geq u_i(A_j - \{y_1^j\})$.

2. Ahora consideremos que j envidia a i, similar al caso anterior, el agente j no envidia al agente i con respecto a O^+ puesto que el agente j elige primero entre i y j; esto es que para cada recurso z_k^i escogido por i, el agente j elige un elemento z_k^j antes que el agente i, tal que uno de los dos tenga poca preferencia por z_k^i, de ese modo se sigue que,

$$u_j(A_j \cap O^+) = \sum_{k=1}^{b} u_j(z_k^j) \geq \sum_{k=1}^{b} u_j(z_k^i) = u_j(A_j \cap O^+) \tag{2.3}$$

Mientras que para los elementos de O^-, para cada elemento w_k^i escogido por el agente i donde $k = 2, 3, ..., a$ hay un elemento w_{k-1}^j escogido por el agente j antes que el agente i donde el agente j tiene poca preferencia por el elemento w_k^i lo cual implica que el agente j no envidia al agente i con respecto a $O^- - \{w_a^j\}$, de donde se sigue que,

$$u_j((A_j \cap O^-) - \{w_a^j\}) \geq \sum_{k=1}^{a-1} u_j(w_k^j) \geq \sum_{k=2}^{a} u_j(w_k^i) \geq \sum_{k=1}^{a} u_j(w_k^i) = u_j(A_i) \tag{2.4}$$

Notar que en la desigualdad (2.4) se cumple que $u_j(w_1^i) \leq 0$, por lo que de que (2.3) y (2.4) se sigue que $u_j(A_j - \{w_a^j\}) \geq u_j(A_i)$.

En cualquier caso, los agentes no se envidian entre sí por más de un elemento. Concluimos que A es libre de envidia de hasta un bien y también lo es la asignación final A^*, ya que la eliminación de elementos ficticios no afecta la utilidad de cada agente. Al analizar el tiempo de ejecución del Algoritmo 1. La línea 1 requiere $O(mn)$ tiempo ya que cada elemento debe ser examinado por todos los agentes. Las líneas 3 y 4 requieren $O(m^2)$ tiempo ya que hay como máximo m iteraciones, y para cada iteración, cada agente considera como máximo m candidatos. Por lo tanto, el tiempo de ejecución total puede estar acotado por $O(\max\{m^2, mn\})$ con lo que se ha probado el resultado. ■

2.1.2. Algoritmo de grafo de envidia generalizada

En la sección anterior el algoritmo 1 está diseñado para utilidades aditivas. Construimos otro algoritmo (Algoritmo 2) que encuentra una asignación EF1 para funciones de utilidad arbitrarias $u_i : 2^O \longrightarrow \mathbb{R}$, no necesariamente aditiva. El algoritmo se basa en una generalización de un algoritmo presentado por [14] para encontrar un EF1 asignación de bienes. Para una asignación A el grafo de bienes $G(A)$ es un grafo dirigido donde los vértices están dados por el conjunto de agentes N y hay un arco de i a j si y sólo si i envidia a j. Para cada ciclo dirigido $C = \{i_1, i_2, ..., i_k\}$ del grafo de envidia $G(A)$ donde i_j envidia a i_{j+1} para cada $j \in [k]$ y $i_{k+1} = i_1$, se puede implementar un intercambio durante el ciclo y definir el asignación resultante A_C de la siguiente manera:

$$A_{i_C} = \begin{cases} A_i, & si \quad i \notin C. \\ \\ A_{i_{j+1}}, & si \quad i = i_j \in C. \end{cases}$$

Un nodo fuente de un grafo dirigido es un vértice sin arcos entrantes, mientras que un sumidero es un vértice sin arcos salientes. Dada una asignación A, la utilidad marginal de un recurso $o \in O/A_i$ a un agente $i \in N$ está definido como $\delta_i(A, o) := u_i(A_i \cup \{o\}) - u_i(A_i)$.

Nuestro algoritmo de grafo de envidia generalizada viene dado de la siguiente manera:

Algorithm 2 Algoritmo de grafo de envidia generalizada

Input: Instancia $I = (N, O, U)$
Output: Una asignación A

```
Inicializar la asignación A_i = ∅ para todo i ∈ N
for o ∈ O do
    Sea N⁺ el conjunto de agentes i que tiene un valor no negativo de
    utilidad marginar δ_i(A, o) ≥ 0.
    if N⁺ ≠ ∅ then
        Elegir un nodo fuente i* ∈ N⁺ en el grafo G(A) inducido por N⁺.
    else
        Elegir un nodo sumidero i* ∈ N⁺ en G(A).
    end if
    Actualizar A_i ← A_{i*} ∪ {o}.
    while G(A) contiene un ciclo C do
        Actualizar A ← A_C.
    end while
end for
```

Ya que el (algoritmo 2) generaliza la función de utilidad en la que está puede ser negativa, veamos el siguiente resultado:

Teorema 2.1.2. *Supongamos que se nos da un acceso de oráculo a las funciones de utilidad $u_i : 2^O \to \mathbb{R}$ de todo agente $i \in N$. Entonces el Algoritmo de grafo de envidia generalizada (Algoritmo 2) encuentra una asignación EF1 en $O(mn^3)$*

Demostración: Se demostrará por inducción que cada vez que un nuevo recurso se asigna y se ejecuta en un ciclo while del algoritmo 2, el grafo de envidia $G(A)$ es acíclico y que la asignación A satisface EF1.

Para nuestra hipótesis inductiva,si la asignación es nula, se sigue el resultado inmediatamente, puesto que no existe envidia para ningún agente y es EF1, mientras que si existe un recurso a repartir entre un número determinado de agentes, como todos los agentes sentirán envidia al agente al que se le asignó el recurso se debería eliminar el único recurso y así seria EF1 y en ambos casos el grafo es acíclico.

Ahora supongamos que se han asignado $k - 1$ recursos y deseamos incluir el recurso k-ésimo $o \in O$.

- Si $N^+ \neq \emptyset$, entonces;
  Sea G^+ el grafo de envidia inducido por N^+, ya que por la hipótesis de inducción el grafo de envidia es acíclico, por tanto, G^+ también lo es gracias a la teoría de grafos y en consecuencia hay al menos un agente i^* que no tiene una arista entrante hacia $i^* \in G+$, por lo que el algoritmo puede dar a i^* el recurso o, puesto que ningún agente envidiaba antes a i^*, la asignación A satisface EF1.

- Si $N^+ = \emptyset$, entonces;
  Notemos que la utilidad marginar sera negativa, es decir, $\delta_i(A, o) < 0$ y como en la parte anterior, gracias a la hipótesis inductiva, el grafo de envidia $G(A)$ es acíclico, por lo que existe al menos un agente i^* al que se le puede dar el recurso o, así puesto que i^* no envidiaba a nadie la asignación A satisface EF1.

Ahora nos centramos en el bucle while del algoritmo mediante el cual se eliminan los ciclos de envidia intercambiando asignaciones a lo largo del ciclo. Después de intercambiar asignaciones durante el ciclo C, observamos que los agentes del ciclo mejoran su utilidad y la envidia hacia cada agente en C sigue limitada a un bien ya que el conjunto de recursos no cambia. Por lo tanto, la asignación A_C resultante todavía satisface EF1, es decir;

Sí suponemos que el ciclo $C = i_1 \to i_2 \to ...i_r \to i_1$ es ciclo dirigido en G, si $A = (A_1, A_2, ..., A_n)$ podemos obtener $A' = A_1', A_2', ..., A_n'$ reasignando los recursos como;

$$A_i' = A_i, \qquad \forall i \in \{i_1, ..., i_r\}$$

Y por lo tanto,

$$A_{i_1}' = A_{i_2}, A_{i_2}' = A_{i_3}, ..., A_{i_r}' = A_{i_1}$$

Notemos que para todos los agentes evaluaron lo que tienen en A' al menos tanto como lo que tienen en A, cabe recalcar que A' usa los mismos recursos que A, solo que se les asigna a distintos propietarios.

Así diremos que G' es el grafo de envidia correspondiente a A', donde se puede observar que el conjunto de aristas entre pares de vértices en N/C no ha cambiado, por lo que cada arista de la forma $i \to i_j$ para $i \in N/C$ y $i_j \in C$, se convierte en G' de la siguiente manera;

$$i \to i_{j-1}$$

Sí $j = 1$ entonces,

$$i \to i_r$$

Por lo tanto, vemos que no se han agregado más aristas de esta forma, mientras que el número de aristas de la forma;

$$i_j \to i \quad \text{con} \quad i_j \in C \quad \text{y} \quad i \in N/C$$

Han disminuido o se mantienen iguales, ya que los agentes en C están estrictamente más conformes, finalmente para $i_j \in C$ el número de aristas de i_j a los vértices en C ha disminuido en la menos 1.

Por lo tanto, al encontrar y eliminar ciclos repetidamente usando el procedimiento anterior y dado que el número de aristas disminuye en cada paso, el proceso terminara y al terminar el grafo G será acíclico y así el tiempo de ejecución del algoritmo puede estar acotado por $O(mn^3)$. ■

2.1.3. Encontrando una asignación EF1 y PO

Ahora pasamos a la siguiente pregunta: si la equidad es alcanzable junto con la eficiencia. En el contexto de la asignación de bienes en la que los agentes tienen una utilidad aditiva no negativa, en [8] se demostró que un resultado que maximiza el bienestar de Nash satisface EF1 y la optimización de Pareto. La pregunta sobre si existe una asignación de PO y EF1 para tareas no está resuelta. Partiendo de una asignación EF1 y encontrando mejoras de Pareto, uno se encuentra con dos desafíos. Primero, las mejoras de Pareto pueden no preservar necesariamente EF1; segundo, encontrar mejoras de Pareto es NP-difícil [2] [13]. Incluso si ignoramos el segundo desafío, la pregunta sobre la existencia de una asignación de PO y EF1 para tareas está abierta. A continuación, mostramos que el problema de encontrar una asignación de PO y EF1 está completamente resuelto para el caso restringido pero importante de dos agentes. Tenga en cuenta que el problema de distribuir un recurso entre dos agentes, que surge en una serie de aplicaciones, como un acuerdo de divorcio y división de tierras, se ha considerado fundamental en la literatura de división justa. De hecho, hay varios trabajos destacados que estudian el caso de dos agentes, que van desde los clásicos [7] y [6] hasta los más recientes[1] [16]. El teorema principal de esta sección se establece de la siguiente manera.

Teorema 2.1.3. *Para dos agentes, siempre existe una asignación Pareto-óptima y EF1 que se puede calcular en un tiempo polinomial.*

Una definición que nos sera útil para probar este resultado es la siguiente:

Definición 2.1.1. *Dada una asignación A, otra asignación A' es una mejora de Pareto de A si, $u_i(A'_i) \geq u_i(A_i)$, para todo $i \in N$ y $u_j(A'_j) > u_j(A_j)$, para algún $j \in N$. Decimos que una asignación es óptima-Pareto(PO) si no hay ninguna asignación que sea una mejora de Pareto de A.*

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
M (winner):	1	-1	2	1	-2	-4	-6
R (loser):	4	-3	6	2	-2	-2	-2
$\lvert u_\ell(o)\rvert \,/\, \lvert u_w(o)\rvert$	4	3	3	2	1	1 / 2	1 / 3

Tabla 2

El algoritmo, que presentaremos a continuación formalmente en el Algoritmo 3, puede verse como una versión discreta de la conocida regla del Ganador ajustado (AW por sus siglas en inglés) [7][6]. Al igual que la regla AW, nuestro algoritmo encuentra una asignación de PO y EF1. A diferencia de AW, que está diseñado para bienes, nuestro algoritmo puede manejar tanto bienes como tareas. Sin pérdida de generalidad, supóngase que no existe ningún ítem para el cual cada agente tenga utilidad 0. El algoritmo comienza dando cada quehacer subjetivo al agente que lo considera un bien o un ítem nulo; asimismo, entrega cada bien subjetivo al agente que lo considera como tal. Entonces, a continuación, asumimos que solo tenemos ítems objetivos, es decir, para cada ítem $o \in O$, o es un recurso tal que,($u_i(o) > 0$ para cada $i \in N$); o de ser el caso el recurso o es una tarea tal que, ($u_i(o) < 0$ para cada $i \in N$). Ahora llamamos a uno de los dos agentes ganador (indicado por w) y otro perdedor (indicado por ℓ).

- Inicialmente, todos los bienes se asignan al ganador y todas las tareas al perdedor.
- Se clasifican los elementos en términos de $|u_\ell(o)| \,/\, |u_w(o)|$ (orden monótono no creciente) de izquierda a derecha, y consideramos la reasignación de los elementos según el orden (desde el elemento más a la izquierda hasta el elemento más a la derecha).
- Al considerar un recurso, lo movemos del ganador al perdedor. Al considerar una tarea, la movemos del perdedor al ganador. Nos detenemos cuando el perdedor no envidia al ganador por más de un elemento.

Algorithm 3 Algoritmo de ganador ajustado generalizado AW

Input: Instancia $I = (N, O, U)$ donde $N = \{w, \ell\}$
Output: Una asignación A

1: Inicializar la asignación $A_i = \emptyset$ para todo $i \in N$
2: Sea $O_w^* = \{o \in O | u_w(o) \geq 0 \ \ y \ \ u_\ell(o) \leq 0\}$ y $O_\ell^* = \{o \in O | u_\ell(o) \geq 0\} \ y \ u_w(o) < 0$.
3: Sea $O^+ = \{o \in O | u_i(o) > 0, \forall i \in N\}$ y $O^- = \{o \in O | u_i(o) < 0, \forall i \in N\}$.
4: Para cada elemento $o \in O^+ \cup O_w^*$, asignar o al agente w. Para cada elemento $o \in O^- \cup O_\ell^*$, asignar el elemento o al agente ℓ.
5: Ordenar los elementos en $O^+ \cup O^- = \{o_1, o_2, ..., o_r\}$ donde $|u_\ell(o_1)| \,/\, |u_w(o_1)| \geq |u_\ell(o_2)| \,/\, |u_w(o_2)| \geq ... \geq |u_\ell(o_r)| \,/\, |u_w(o_r)|$.
6: Sea $t = 1$.
7: **while** El agente ℓ envidia al agente w por más de un elemento **do**
8:     **if** $o_y \in O^+$ **then**
9:         Sea $A_w = A_w - \{o_t\}$ y $A_\ell = A_\ell \cup \{o_t\}$
10:     **elseif** $o_t \in O^-$
11:         $A_w = A_w \cup \{o_t\}$ y $A_\ell = A_\ell - \{o_t\}$
12:     **end if**
13:     Actualizar $t = t + 1$
14: **end while**

De donde obtenemos el siguiente lema que nos servirá para probar nuestro **teorema 2.1.3**.

Lema 2.1.1. *Durante la ejecución del Algoritmo 3, la asignación A es Pareto-óptima en cualquier punto después de la línea 4.*

Demostración: Se puede ver fácilmente que la asignación justo después de la línea 4 es Pareto-óptima. Por lo tanto, consideremos algún paso de tiempo después de que el algoritmo ingrese al ciclo while de la Línea 8. Supongamos por contradicción de que A' es una mejora de Pareto de A. Asumimos sin pérdida de generalidad que todos los elementos en O_w^* permanecen asignados a w bajo A' porque transferir un elemento en O_w^* de w a ℓ no mejora ni la utilidad de w ni la de ℓ. Asimismo, asumimos que todos los elementos en O_ℓ^* permanecen asignados a ℓ bajo A'.

A continuación, llamamos a cada $o \in O^+$ un recurso y a cada $o \in O^-$ una tarea. Para cada $i, j \in \{w, \ell\}$ con $i \neq j$, sea

- G_{ii} el conjunto de recursos en $A_i \cap A'_i$;
- C_{ii} el conjunto de tareas en $A_i \cap A'_i$;
- G_{ij} el conjunto de recursos en $A_i \cap A'_j$; y
- C_{ij} el conjunto de tareas en $A_i \cap A'_j$.

Consideremos primero el caso cuando en A, el ganador tiene una utilidad que es al menos tan alta como en A', mientras que el perdedor está estrictamente mejor. Teniendo en cuenta el hecho de que los conjuntos de recursos G_{ww} y $G_{\ell\ell}$ y el conjunto de tareas C_{ww} y $C_{\ell\ell}$, Se asignan al mismo agente en ambas asignaciones, esto significa lo siguiente:

$$u_w(G_{\ell w}) + u_w(C_{\ell w}) - u_w(G_{w\ell}) - u_w(C_{w\ell}) \geq 0 \tag{2.5}$$

$$u_\ell(G_{w\ell}) + u_\ell(C_{w\ell}) - u_\ell(G_{\ell w}) - u_\ell(C_{\ell w}) > 0 \tag{2.6}$$

Podemos observar que ahora el algoritmo consideró todos los elementos en $G_{\ell w}$ y $C_{w\ell}$ antes que los elementos en $G_{w\ell}$ y $C_{\ell w}$ en el ordenamiento. Cabe recalcar que todos los recursos se asignan inicialmente al ganador(w), $G_{\ell w} \subseteq A_\ell$ y $G_{w\ell} \subseteq A_w$. Así, los recursos en $G_{\ell w}$ son los que se transfieren del ganador w al perdedor ℓ en el bucle while de la línea 8, mientras que los recursos en $G_{w\ell}$ son los que permanecen en la asignación del ganador. Del mismo modo, recordar que todas las tareas se asignan inicialmente al perdedor, $C_{w\ell} \subseteq A_w$ y $C_{\ell w} \subseteq A_\ell$. Así, las tareas en $C_{w\ell}$ son los que se transfieren del perdedor ℓ al ganador w, mientras que las tareas en $C_{\ell w}$ son los que permanecen en la asignación del perdedor.

Ahora, sea α tal que;

$$\max_{o\in G_{w\ell}\cup C_{\ell w}} |u_\ell(o)| \,/\, |u_w(o)| \leq \alpha \leq \min_{o\in G_{\ell w}\cup C_{w\ell}} |u_\ell(o)| \,/\, |u_w(o)|$$

Lo cual implica las siguientes desigualdades,

$$u_\ell(G_{w\ell}) \leq \alpha u_w(G_{w\ell});$$

$$u_\ell(G_{\ell w}) \geq \alpha u_w(G_{\ell w});$$

$$-u_\ell(C_{w\ell}) \geq -\alpha u_w(C_{w\ell});$$

$$-u_\ell(C_{\ell w}) \leq -\alpha u_w(C_{\ell w})$$

Que gracias a la desigualdad (2.6), nos queda,

$$0 < u_\ell(G_{w\ell})+u_\ell(C_{w\ell})-u_\ell(G_{\ell w})-u_\ell(C_{\ell w}) \leq -\alpha(u_w(G_{\ell w})+u_w(C_{\ell w})-u_w(G_{\ell w}-u_w(C_{w\ell})) \leq 0$$

Lo cual es una contradicción, pues la última desigualdad contradice lo que se sigue de la desigualdad (2.5) y del hecho de que α no es negativo. Similarmente se aplica cuando en A, el perdedor tiene una utilidad que es al menos tan alta como en A', mientras que el ganador está estrictamente mejor. ■

Ahora podemos demostrar nuestro teorema principal de esta sección.

Demostración: **Teorema 2.1.3** Probaremos que la asignación final A del Algoritmo 3 satisface EF1 que junto con el **Lema 2.1.1**, esto prueba lo que deseamos.

Ahora, veamos que en la asignación final A, un agente como mucho envidia al otro: si el perdedor todavía envidia al ganador y el ganador también envidia al perdedor, entonces el intercambio de asignaciones resultaría en una mejora de Pareto, contradiciendo el **Lema 2.1.1**. Así , si el perdedor envidia al ganador en A, el ganador no envidia al perdedor, lo que implica que A es EF1.

Considere cuando en A, el perdedor no envidia al ganador pero el ganador envidia al perdedor. Sea A' la asignación previa justo antes de la transferencia final en el bucle while de la Línea 8. Sea $W = A'_w \cap A_w$ y

$L = A'_\ell \cap A_\ell$. Es decir, W es el conjunto de elementos en el paquete del ganador y L esel conjunto de elementos del perdedor respectivamente, excluyendo el elemento transferido en A' y A. Por construcción, el perdedor envidia al ganador por más de un ítem enA', lo que implica $u_\ell(L) < u_\ell(W)$. Supongamos por contradicción que el ganador envidia al perdedor por más de un elemento en A, lo que implica $u_w(W) < u_w(L)$.

- Si h es el último recurso que se ha movido del ganador al perdedor, se asigna W a ℓ y $L \cup \{g\}$ a w sería una mejora de Pareto de A', lo cual es una contradicción.
- Si c es la última tarea que se ha movido del perdedor al ganador, entonces se le asigna $W \cup \{c\}$ a ℓ y L a w sería una mejora de Pareto de A', lo cual es una contradicción.

Por lo tanto, el ganador no envidia al perdedor por más de un elemento en A, y concluimos que A es EF1.

Nos queda por analizar el tiempo de ejecución del algoritmo. Primero, los elementos se pueden ordenar en tiempo $O(m * log(m))$. El proceso de ajuste lleva $O(m^2)$ de tiempo. Cada iteración verifica si la asignación es EF1 desde la perspectiva del perdedor, lo que requiere como máximo m comparaciones de utilidades y hay como máximo m iteraciones. Por tanto, el número de operaciones está acotado por $O(m^2)$. ■

Ejemplo 2.1.2. *Consideremos dos agentes, M y R, y cinco elementos con las utilidades aditivas especificadas en la Tabla 2, donde los elementos* $1, 3$ *y* 4 *corresponden a los recursos y los elemenros* $2, 5, 6$ *y* 7 *corresponden a las tareas. El AW generalizado inicialmente asigna los recursos a M y las tareas a R. Luego, transfiere el primer recurso de M a R y mueve la segunda tarea de R a M. Después de mover el tercer recurso de M a R, R deja de sentir envidia por un elemento. Por lo tanto, la asignación final otorga los elementos* 2 *y* 4 *a M y el resto a R.*

Una pregunta natural es si existen asignaciones de PO y EF1 para tres o más agentes; dejamos esto como una interesante pregunta abierta. Hacemos notar que la optimización de Pareto por sí misma es fácil de lograr en tiempo $O(nm)$, tomando una permutación de agentes y aplicando una variante de "dictadura en serie".

Proposición 2.1.2. *Una asignación óptima de Pareto se puede calcular en tiempo lineal.*

Para verificar la demostración a detalle de la **Proposición 2.1.2** revisar el siguiente articulo [12].

Capítulo 3

Resultados, conclusiones y recomendaciones

3.1. Resultados

Pudimos ver que siempre existe una asignación EF1 para recursos y tareas subjetivos. Si se debilita EF1 a PROP1, se puede lograr un requisito adicional además de la equidad, la conectividad. En este capitulo, consideraremos una situación en la que los elementos se colocan en una ruta y cada agente desea una asignación conectada de la ruta. Encontrar un conjunto de elementos conectados es importante en muchos escenarios. Por ejemplo, los elementos pueden ser un conjunto de habitaciones en un corredor y los agentes pueden ser grupos de investigación donde cada grupo de investigación quiere obtener habitaciones adyacentes revisar [4] y [5].

Se mostrara que existe una asignación PROP1 conectada y que se puede encontrar de manera eficiente. A continuación, asumimos que el camino está dado por una secuencia de elementos $(o_1, o_2, ..., o_m)$. Formalmente, decimos que una asignación A es conexa si para cada agente $i \in N$, A_i es conexo en el camino $(o_1, o_2, ..., o_m)$. Para ello consideremos una noción un poco más estricta de PROP1: Una asignación conectada A es PROP1_{outer}, si para cada $i \in N$ se tiene que;

- El agente i recibe una asignación de utilidad de al menos su parte justa proporcional, es decir, $u_i(A_i) \geq \frac{u_i(O)}{n}$; o

- $u_i(A_i) + u_i(o) \geq u_i(O)/n$ para algún $o \in O - \{A_i\}$ tal que $A_i \cup \{o\}$ están conectados.

- $u_i(A_i) - u_i(o) \geq u_i(O)/n$ para algún $o \in A_i$ tal que $A_i - \{o\}$ están conectados.

Primero se demostrara el resultado para un caso de corte de torta [7] y [17] que es de interés independiente. En lo que sigue, un pastel mixto es el intervalo $[0, m]$. Cada agente $i \in N$ tiene una función de densidad de valor $\hat{u}_i$, que asigna un subintervalo del pastel a un valor real, donde i tiene una utilidad uniforme $u_i(o_j)$ para el intervalo $[j-1, j]$ para cada $j \in [m]$. La parte justa proporcional del agente i para un pastel mixto $[0, m]$ viene dada por $\hat{u}_i([0, m])/n$. Una asignación contigua de un pastel mixto asigna a cada agente un subintervalo disjunto del pastel en el que la unión de los intervalos es igual al pastel completo $[0, m]$; satisface la proporcionalidad si cada agente i obtiene un intervalo de utilidad que es al menos su parte proporcional justa.

Teorema 3.1.1. *Existe una asignación proporcional contigua de un pastel mixto y se puede calcular en tiempo polinomial.*

Demostración: Sea N^+ el conjunto de agentes con utilidad total estrictamente positiva para O. Se Combina el algoritmos de cuchillo móvil(moving-knife) para recursos y tareas de la siguiente manera. Primero, si hay un agente que tiene una participación justa proporcional positiva $N^+ \neq \emptyset$, aplicamos el algoritmo de cuchillo móvil solo a los agentes en N^+. Nuestro algoritmo mueve un cuchillo de izquierda a derecha y los agentes vociferan cada vez que la parte izquierda del pastel tiene una utilidad exactamente igual a la parte justa proporcional. Al primer agente que vocifera se le asigna el paquete izquierdo y el algoritmo se repite en la instancia restante. En segundo lugar, si ningún agente tiene una parte justa proporcional positiva, nuestro algoritmo mueve un cuchillo de derecha a izquierda, y los agentes vociferan cada vez que la parte izquierda del pastel tiene una utilidad proporcionalmente justa. Nuevamente, al primer agente que vocifera se le asigna el paquete izquierdo y el algoritmo se repite en la instancia restante. Para una asignación A y agentes $N' \subseteq N$, definimos $A|_{N'} : N' \longrightarrow 2^O$ como restricción de A a N', es decir, $A|_{N'}(i) = A_i$ para cada agente $i \in N'$.

Algorithm 4 Algoritmo de cuchillo móvil generalizado

Input: Un sub-intervalo $[\ell, r]$, conjunto de agentes N', funciones de utilidad $\hat{u}_i$ para cada $i \in N'$.

Output: Una asignación $\hat{A}$ de una pastel mixto $[\ell, r]$ a N'

Inicializar la asignación $\hat{A}_i = \emptyset$ para cada $i \in N'$.
Sea $N^+ = \{i \in N' | \hat{u}_i([\ell, r]) > 0\}$.
if $N^+ \neq \emptyset$ **then**
  if $|N^+| = 1$ **then**
    Asignar $[\ell, r]$ al único agente en N^+.
  else
    Sea x_i el punto mínimo donde $\hat{u}_i([\ell, x_i]) = \hat{u}_i([\ell, r])/|N'|$ para $i \in N^+$.
    Encontrar el agente j con mínimo x_j entre todos los agentes en N^+.
    Retornar $\hat{A}$ donde $\hat{A}_j = [\ell, x_j]$ y $A|_{N'-\{j\}} = \mathcal{A}([x_j, r], N' - \{j\}, (\hat{u}_i)_{i \in N'-\{j\}})$.
  end if
else
  Sea x_i el punto máximo donde $hatu_i([\ell, x_i]) = \hat{u}_i([\ell, r])/n$ para $i \in N'$.
  Encuentra el agente j con máximo x_j entre todos los agentes en N'.
  Retornar $\hat{A}$ donde $\hat{A}_j = [\ell, x_j]$ y $A|_{N'-\{j\}} = \mathcal{A}([x_j, r], N' - \{j\}, (\hat{u}_i)_{i \in N'-\{j\}})$.
end if

El algoritmo 4 formaliza las ideas dichas en los párrafos anteriores. Para probar su exactitud, demostraremos por inducción sobre el número de agentes N', que la asignación de una torta mixta $[\ell, r]$ producida por $\mathcal{A}$ satisface las siguientes condiciones:

- si $N \neq \emptyset$, entonces cada agente en N^+ recibe un intervalo de utilidad que es al menos su parte justa proporcional $\hat{u}_i([\ell, r])/|N'|$ y cada agente, que no está en N^+ recibe una pieza vacía.
- si $N = \emptyset$, entonces cada agente recibe un intervalo de utilidad al menos su parte justa proporcional $\hat{u}_i([\ell, r])/|N'|$.

Claramente el resultado se sigue cuando $|N'| = 1$. Supongamos que $\mathcal{A}$ nos da una asignación proporcional de un pastel mixto con las propiedades deseadas cuando $|N'| = k - 1$; lo probaremos para $|N'| = k$. Suponga que algún agente tiene una participación justa proporcional positiva, es decir, $N^+ \neq \emptyset$. Notemos que cada agente i que no está en N^+ tiene una parte justa proporcional no positiva y no recibe nada; por lo tanto, basta con mostrar que los agentes en N^+ reciben paquetes de utilidad que son al menos su parte justa proporcional. Si $|N^+| = 1$, la afirmación es trivial; por lo tanto, supongamos lo contrario. Podemos ver que claramente, el agente j recibe un intervalo de utilidad que es al menos su parte justa proporcional. Además, todos los demás agentes en N^+ tienen utilidad como máximo en sus partes proporcionales justas para la pieza izquierda $[\ell, x_j]$. De hecho, si hay un agente $i' \in N^+$ cuya utilidad por la pieza izquierda $[\ell, x_j]$ es mayor que su parte justa proporcional, es decir, $\hat{u}_{i'}([\ell, x_j])/k$, entonces i' habría vociferado cuando el cuchillo llega antes de x_j por la continuidad de $\hat{u}_{i'}$, es decir, $x_{i'} < x_j$, contradiciendo la minimalidad de x_j. Así, los agentes restantes en N^+ tienen al menos $(k-1) * \hat{u}_i([\ell, r])/k$ utilidad para el resto del pastel $[x_j, r]$; por lo tanto, por la hipótesis de inducción, cada agente en N^+ obtiene un intervalo de utilidad que es al menos su parte justa proporcional, y cada uno de los agentes restantes obtiene una pieza vacía.

Ahora supongamos que ningún agente tiene una participación justa proporcional positiva. Nuevamente, si hay un agente i' cuya utilidad para la pieza izquierda [, xj] es mayor que su parte justa proporcional $\hat{u}_{i'}([\ell, r])/k$, entonces i' habría vociferado cuando el cuchillo llega antes de x_j por la continuidad de la función $\hat{u}_{i'}$, es decir, $x_{i'} > x_j$, contradiciendo la maximalidad de x_j. Por lo tanto, todos los agentes restantes tienen una utilidad de al menos $(k-1) * \hat{u}_i([\ell, r])/k$ para el resto del pastel $[x_j, r]$, y por lo tanto, según la hipótesis inductiva, cada el agente obtiene un intervalo de utilidad que es al menos su parte proporcional justa $\hat{u}_i([\ell, r])/k$. Se puede probar fácilmente que el Algoritmo 4 se ejecuta en tiempo polinomial. ■

El teorema establecido anteriormente también se aplica a un modelo general de corte de torta en el que la información sobre la función de utilidad del agente durante un intervalo puede inferirse mediante una serie de consultas. Observamos que, en contraste con la proporcionalidad, la existencia de una relación contigua La asignación libre de envidia de un pastel mixto sigue siendo esquiva: se sabe que existe solo cuando el número n de agentes es cuatro o un número primo [15], [18]. A continuación, mostramos cómo una asignación proporcional fraccionaria (una asignación que logra la proporcionalidad pero trata los artículos como divisibles) puede usarse para lograr una división PROP1 contigua de artículos indivisibles.

Teorema 3.1.2. *Siempre existe una asignación PROP1$_{outer}$ conectada a una ruta y se puede calcular en tiempo polinomial.*

Demostración: Consideremos un camino $(o_1, o_2, ..., o_m)$, un pastel mixto $[0, m]$ y que cada agente tenga una función de dencidad $\hat{u}_i$, donde i tiene una utilidad uniforme $u_i(o_j)$ para el intervalo $[j-1, j]$, para cada $j \in [m]$. Sabemos que esta instancia admite una asignación contigua y proporcional $\hat{A}$ gracias al **Teorema 3.1.1**. Supongamos sin pérdida de generalidad que bajo la asignación $\hat{A}$, los agentes $1, 2..., n$ reciben los paquetes $1er, 2nd, ..., n - simo$ de izquierda a derecha. Es decir, cada agente $i = 1, 2..., n$ recibe el subintervalo $[x_{i-1}, x_i]$ de la torta mixta, donde, $0 = x_0 \leq x_1 \leq ... \leq x_{n-1} \leq x_n = m$. Sin pérdida de generalidad, también supongamos que ningún agente se le da la cesta vacía, bajo esta asignación fraccionaria, es decir, $x_{i-1} < x_i$ para cada $i \in N$.

De izquierda a derecha, mostraremos cómo asignar cada elemento o_j para cada $j \in O$ para construir una asignación integral. Si el elemento o_j está totalmente contenido en la cesta de algún agente, es decir, $x_{i-1} \leq j - 1 \leq j \leq x_i$ para algun $i \in N$, entonces asignamos cada elemento o_j al agente i. Si no (es decir, el elemento o_j está en el límite), lo asignamos de acuerdo con la preferencia de los agentes más a la izquierda o más a la derecha. Formalmente, supongamos que, $j-1 \leq x_\ell \leq x_{\ell+1} \leq ... \leq x_r \leq j$ tal que $x_\ell = \min\{x_i | x_i \geq j-1\}$ y $x_r = \max\{x_i | x_i \leq j\}$. Realizamos lo siguiente presentado a continuación:

1. Si dos agentes ℓ y r no están de acuerdo en el signo de o_j, es decir, $\min\{u_\ell(o_j), u_r(o_j)\} < 0 < \max\{u_\ell(o_j), u_r(o_j)\}$, luego entregamos el elemento o_j al agente $i \in \{\ell, r\}$ que tiene una utilidad positiva para ello.

2. Si dos agentes ℓ y r están de acuerdo con el signo de o_j, es decir, $\min\{u_\ell(o_j), u_r(o_j)\} \geq 0$ o $\max\{u_\ell(o_j), u_r(o_j)\} < 0$, luego asignamos el elemento o_j de tal manera que:

- El agente izquierdo ℓ toma o_j si ambos agentes tienen una utilidad no negativa (es decir, $\text{mín}\{u_\ell(o_j), u_r(o_j)\} \geq 0$).
- El agente derecho r toma o_j si ambos agentes tienen una utilidad negativa (es decir, $\text{máx}\{u_\ell(o_j), u_r(o_j)\} < 0$).

Tengamos en cuenta que, si bajo la división fraccionaria proporcional, el agente i obtiene una fracción de un solo elemento y hay otros dos agentes $i-1$ e $i+1$ a la izquierda y a la derecha que obtienen una fracción del mismo elemento, el agente i no obtiene nada bajo nuestra asignación final.

La asignación integral resultante A es PROP1$_{outer}$. Para ver esto, tomemos cualquier agente $i \in N$, se puede ver claramente que, cuando una de las posiciones del cuchillo x_{i-1} y x_i es integral, el paquete satisface PROP1$_{outer}$. Además, si $[x_{i-1}, x_j] \subseteq [j-1, j]$ para algún $j \in [m]$, el agente i obtiene una utilidad $1/n$ al recibir el artículo o_j o la cesta vacía. Así, supongamos lo contrario, es decir, $x_{i-1}, x_i \notin \{0, 1, ..., m\}$ y $x_i - x_{i-1} > 1$. Mostraremos que dicho agente obtiene una utilidad $1/n$ al recibir el artículo junto a su paquete o al eliminar el artículo más a la izquierda de su paquete. Sean o_ℓ y o_r los elementos de contorno izquierdo y derecho donde, $x_{i-1} \in (r-1, r)$ y $x_i \in (\ell-1, \ell)$, tengamos en cuenta que tenemos $\{o_{\ell+1}, o_{\ell+2}, ..., o_{r-1}\} \subseteq A_i)$. Donde consideramos los siguiente cuatro casos:

- Tanto o_ℓ y o_r son recursos o elementos nulos para i, es decir, $\text{mín}\{u_i(o_\ell), u_i(o_r)\} \geq 0$. En este caso, el agente i recibe al menos o_r. Así, si $o_\ell \in A_i$, el agente i obtiene una utilidad $1/n$. Si no, el agente i obtiene la utilidad $1/n$ al recibir el artículo o_ℓ.
- Tanto o_ℓ y o_r son tareas para i, es decir, $\text{máx}\{u_i(o_\ell), u_i(o_r)\} < 0$. En este caso, el agente i no recibe o_r. Así, si $o_\ell \notin A_i$, el agente i obtiene una utilidad $1/n$. Si no, el agente i obtiene la utilidad $1/n$ al eliminar el elemento o_ℓ.
- El elemento o_ℓ es un recurso o elemento nulo pero o_r es una tarea para i, es decir, $u_i(o_\ell) \geq 0$ y $u_i(o_r) < 0$. En este caso, el agente i no recibe o_r. Así, si $o_\ell \in A_i$, el agente i obtiene una utilidad $1/n$. Si no, el agente i obtiene la utilidad $1/n$ al recibir el artículo o_ℓ.

- El elemento o_ℓ es una tarea elemento nulo pero o_r es un recurso o elemento nulo para i, es decir, $u_i(o_\ell) < 0$ y $u_i(o_r) \geq 0$. En este caso el agente i recibe al menos al elemento o_r.Así, si $o_\ell \notin A_i$, el agente i obtiene una utilidad $1/n$. Si no, el agente i obtiene la utilidad $1/n$ al eliminar el elemento o_ℓ.

Se concluye que A es una asignación PROP1_{outer} conectada. Por el **Teorema 3.1.1**, es inmediato ver que se puede calcular una asignación PROP1_{outer} conectada en tiempo polinomial. ■

3.2. Conclusiones y recomendaciones

En el presente trabajo, hemos analizado y realizado formalmente los resultados de la asignación justa cuando los elementos indivisibles son una combinación de recursos subjetivos y tareas. Este trabajo deja una interesante línea de investigación con muchos problemas abiertos para futuras exploraciones.

Asignaciones EF1 para funciones de utilidad general. Mostramos que para las utilidades aditivas generales, existe una asignación EF1 y se puede calcular de manera eficiente mediante el procedimiento de doble turno rotativo(DOUBLE Round Robin). Una pregunta abierta como resultado del presente trabajo puede ser la existencia de una asignación EF1 bajo utilidades arbitrarias no monótonas; Otra cuestión abierta es la existencia de dotaciones de PO y EF1 en nuestro medio. Si bien nuestro trabajo establece la existencia de tales asignaciones para dos agentes con utilidades aditivas, el problema para un número arbitrario de agentes está abierto, incluso para el entorno restringido en el que los elementos son tareas. Aquí, discutimos algunos enfoques fallidos mientras intentamos probar que existe un resultado EF1 y PO para las tareas.

- Recordemos que analizamos resultados en los cuales resolvían problemas de PO y EF1 para bienes. Pero maximizar el producto de las utilidades parece fútil porque el objetivo es positivo o negativo según la paridad del número de agentes.

- Otro enfoque potencial para las tareas es minimizar el producto de las desutilidades; si el producto es cero, podemos identificar un conjunto más grande de agentes para los cuales el producto es positivo. Entonces podemos aplicar la solución a este conjunto de agentes. Tal enfoque no otorga una garantía EF1. Para ver esto, consideremos el caso de dos agentes y cuatro tareas idénticas de utilidad -1. En ese caso, el resultado que asigna al menos un artículo a cada agente y minimiza el producto de las desutilidades es uno en el que un agente obtiene una tarea y el otro obtiene tres tareas. El resultado no satisface EF1 a pesar de la existencia de un resultado equilibrado que asigna un número igual de tareas a cada agente.

- Uno de los resultados que funciona bien para la equidad en el caso de tareas divisibles es la regla que, entre todas las asignaciones de PO, maximiza el producto de las desutilidades de aquellos agentes que obtienen una desutilidad distinta de cero. Sin embargo, existe un ejemplo sencillo con dos agentes tal que maximizar el producto de las desutilidades sujetas a PO no otorga garantía EF1.

Asignaciones Round Robin Share(RRS) y PO. Observamos que, si bien la relación entre la optimización de Pareto y la mayoría de las nociones de equidad aún no está clara, [11] propusieron un concepto de equidad llamado round-robin share (RRS) que se puede lograr junto con la optimización de Pareto. En nuestro contexto, RRS se puede formalizar de la siguiente manera. Dada una instancia $I = (N, O, U)$, considere la secuencia rotativa en la que todos los agentes tienen la misma utilidad que el agente i. En ese caso, la utilidad mínima alcanzada por cualquiera de los agentes es $RRS_i(I)$. Una asignación satisface RRS si cada agente i obtiene una utilidad de al menos $RRS_i(I)$. Entonces nos preguntamos cuál es la complejidad computacional de encontrar una asignación que satisfaga ambas propiedades.

Asignaciones Libres de envidia hasta el objeto menos valorado (EFX). Hay otros conceptos de equidad que podrían examinarse tanto desde la perspectiva de la existencia como de la complejidad, en particular la ausencia de envidia hasta el elemento menos valorado (EFX)[9]. En nuestro entorno, se puede definir una asignación A como EFX si para cualquier par de agentes i y j, el agente i no envidia al agente j, o se cumplen las dos condiciones siguientes:

- $\forall o \in A_i$ tal que $u_i(A_i)/\{o\}) > u_i(A_i) : u_i(A_i)/\{o\}) \geq u_i(A_j)$; y
- $\forall o \in A_j$ tal que $u_i(A_j/\{o\}) < u_i(A_i) : u_i(A_i) \geq u_i(A_j/\{o\})$.

Es decir, la envidia de i hacia j se puede eliminar quitando el recurso menos valioso de i del paquete de j o eliminando la tarea favorita de i del paquete de i. En [11] se menciona el siguiente problema "enigmático": ¿existe una asignación de EFX para bienes? lo cual sería interesante investigar la misma pregunta para bienes/tareas subjetivas u objetivas bajo la utilidad aditiva.

Referencias bibliográficas

[1] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 545–562, 2017.

[2] Haris Aziz, Peter Biro, Jérôme Lang, Julien Lesca, and Jérôme Monnot. Optimal reallocation under additive and ordinal preferences. In *2016 International Conference on Autonomous Agents And Multiagent Systems*, pages 402–410, 2016.

[3] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores.

[4] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S Zwicker. Almost envy-free allocations with connected bundles. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[5] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 135–141, 2017.

[6] Steven Brams and Alan D Taylor. A procedure for divorce settlement. *Mediation Quarterly*, 13(3):191–205, 1996.

[7] Steven J Brams, Steven John Brams, and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[8] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 305–322, 2016.

[9] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[10] Ulle Endriss. Lecture notes on fair division. 2010.

[11] Brandon Fain, Kamesh Munagala, and Nisarg Shah. Fair allocation of indivisible public goods. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 575–592, 2018.

[12] Ayumi Igarashi and Dominik Peters. Pareto-optimal allocation of indivisible goods with connectivity constraints. 2019.

[13] Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *International Conference on Algorithmic Decision Theory*, pages 98–110. Springer, 2009.

[14] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131, 2004.

[15] Frédéric Meunier and Shira Zerbib. Envy-free cake division without assuming the players prefer nonempty pieces. *Israel Journal of Mathematics*, 234(2):907–925, 2019.

[16] Benjamin Plaut and Tim Roughgarde. Almost envy-freeness with general valuations. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2584–2603. SIAM, 2018.

[17] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can.* AK Peters/CRC Press, 1998.

[18] Erel Segal-Halevi. Fairly dividing a cake after some parts were burnt in the oven. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1276–1284, 2018.