UNIVERSIDAD DE MURCIA, FACULTAD DE MATEMÁTICAS

DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

TRABAJO FIN DE GRADO

El problema de Haplotipaje de Poblaciones de la Parsimonia Pura

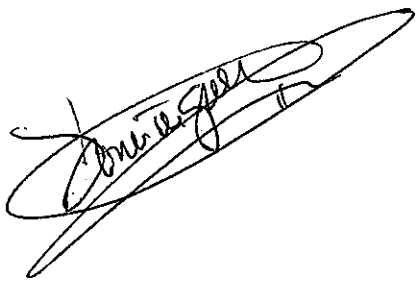
Contextualización y recorrido cronológico a través de distintos modelos propuestos para su resolución.

Realizado por:
Concepción Domínguez Sánchez
Dirigido por:
Alfredo Marín Pérez

Julio 2016

CONCEPCIÓN DOMÍNGUEZ SÁNCHEZ, autora del TFG "El Problema de Haplotipaje de la Parsimonia Pura: Recorrido Cronológico a través de distintos problemas propuestos para su resolución", bajo la tutela del profesor ALFREDO MARÍN PÉREZ, declara que el trabajo que presenta es original, en el sentido de que se ha puesto el mayor empeño en citar debidamente todas las fuentes utilizadas.

En Murcia, a 12 de julio de 2016,



Índice general

Resumen

El Proyecto Genoma Humano (PGH) es una iniciativa que surge en 1990 con la finalidad de secuenciar el genoma completo de nuestra especie, es decir, descifrar el ADN contenido en cada célula de un ser humano. Nuestro ADN está organizado en pares de cromosomas, de los cuales una copia se hereda del padre, y la otra de la madre, y se puede describir mediante una secuencia de cuatro bases que es, en su mayor parte, igual para todos los individuos. El PGH finalizó en 2003, llegándose a descifrar la secuencia de más de tres mil millones de bases y abriendo la puerta a nuevas investigaciones, aún en proceso, que tienen como objetivo establecer las diferencias entre el ADN de dos individuos. Las diferencias entre individuos debidas a la alteración de una única base de la secuencia se conocen con el nombre de SNP (del inglés, Single Nucleotide Polymorphism), y son, junto con los procesos de recombinación, la forma predominante de variación genética, pudiendo encontrarse un SNP en un cromosoma de media cada mil bases. Conocer estas diferencias es vital en el diagnóstico y tratamiento de enfermedades géneticas, así como en el desarrollo de fármacos y en el diseño de drogas. Su comprensión permitirá, además, ampliar nuestros conocimientos sobre la evolución y seguir avanzando hacia la comprensión de procesos celulares básicos, como la síntesis de proteínas y su función.

Los procesos de obtención de la secuencia de bases heredada de cada progenitor son muy costosos en tiempo y dinero, pero existen procesos más accesibles que permiten obtener información sobre los cromosomas. En concreto, dan información acerca de qué bases hay en cada alelo (i.e., en cada posición concreta de un cromosoma), indicando si un individuo es *homocigótico* (si tiene las mismas bases en dicho alelo), o *heterocigótico* (si sus progenitores han aportado bases distintas). Esta información es conocida como el genotipo de un individuo. Sin embargo, cuando el individuo es heterocigótico para un alelo, el genotipo no permite conocer qué base ha aportado cada progenitor.

En este contexto surgen, a finales del s. XX, los problemas de haplotipaje, que tienen la finalidad de, dado el genotipo de un individuo, averiguar la secuencia de bases que ha aportado cada progenitor, llamada haplotipo. El proceso de haplotipaje de una población puede llevarse a cabo persiguiendo distintas motivaciones biológicas. Uno de los criterios ampliamente aceptados es el de la Parsimonia Pura, que tiene el objetivo de encontrar el menor número posible de haplotipos que pueden dar lugar a una población dada. Una de las motivaciones biológicas que llevan a adoptar este criterio es que el número de haplotipos que se observan en la naturaleza es muy inferior al que podría deducirse a partir de la cantidad de SNP con que cuenta el ADN humano, que permite una enorme variación. Además, si partimos de la base de que descendemos de un número reducido de ancestros, resulta lógico pensar que sus haplotipos son los mismos que poseemos hoy en día (excluyendo los procesos de recombinación genética y las mutaciones).

En este trabajo hemos realizado un estudio de distintos modelos propuestos en las dos últimas décadas para abordar el problema de Haplotipaje de Poblaciones mediante el criterio de la Parsimonia Pura.

En el primer capítulo del trabajo incluimos contenidos que necesitaremos dominar a lo largo de los capítulos siguientes. En el primer apartado damos la definición de varios conceptos básicos de Teoría de Grafos, necesarios para comprender el modelo teórico que detallamos en el último capítulo. Asimismo, también incluimos una sección con la defi-

nición de emparejamiento en un grafo y algunos teoremas y resultados que nos permiten encontrar un emparejamiento máximo en un grafo, lo cual también nos será de utilidad en el último capítulo. Finalmente, incorporamos definiciones de modelos de optimización lineal y de optimización entera, ya que los modelos que estudiamos en el capítulo 3 son todos de este tipo.

El segundo capítulo es una introducción a los problemas de haplotipaje. En él se detalla el origen del problema, dando las motivaciones biológicas que provocan su estudio. Para ello, habremos de definir, desde un punto de vista biológico, todas las nociones que utilizaremos con frecuencia durante la explicación de los modelos. También se incluye en este capítulo una sección en la que se definen los dos tipos de problemas de haplotipaje existentes, sus objetivos y algunos de los criterios más utilizados en su resolución. Este capítulo constituye, por lo tanto, una contextualización del problema de Haplotipaje de la Pura Parsimonia (o HPP), objeto central del trabajo.

En el tercer capítulo se explican con detalle los planteamientos y formulaciones de cuatro modelos propuestos para abordar el problema HPP, todos ellos de optimización lineal entera. El capítulo comienza con una sección en la que se incluye la notación matemática que emplearemos posteriormente en la descripción de los modelos, así como algunas definiciones. El primero de los modelos recogidos en el trabajo es el Modelo TIP, que fue propuesto por Gusfield en 2003, siendo el primero de los que existen formulados mediante optimización lineal entera. Esta formulación incluye la creación de una variable de decisión por cada haplotipo que puede haber dado lugar a uno de los genotipos de la población, para posteriormente tratar de minimizar el número de haplotipos, intentando escoger el mayor número de haplotipos que participen en la generación de más de un genotipo. Debido a que el número de haplotipos que pueden dar lugar a un genotipo crece de manera exponencial con respecto al número de alelos heterocigóticos del mismo, las variables de la formulación crecen exponencialmente con respecto al tamaño del problema. Gusfield propone en el mismo artículo una manera de reducir el elevado número de variables, dando lugar al Modelo RTIP, aunque los estudios llevados a cabo muestran que el modelo RTIP no resulta práctico para problemas de más de 30 SNP y 50 individuos.

El segundo modelo estudiado es el Modelo SC (de *Set Covering*, en inglés). Este modelo fue propuesto en 2009 por Lancia y Serafini, y está basado en una condición de cubrimiento de conjuntos que se demuestra que cumplen los haplotipos en cualquier solución factible del HPP. Esta condición establece que, para una posición heterocigótica de un genotipo, han de existir en cualquier solución factible dos haplotipos que puedan haber generado dicho genotipo, cada uno de los cuales con una de las bases que posee el genotipo en dicha posición. Al ser una condición necesaria pero no suficiente para un conjunto solución de haplotipos, esta condición constituye una relajación del HPP, ya que, como veremos posteriormente, existen soluciones óptimas que cumplen esta condición y que, sin embargo, son infactibles para el HPP. Para eliminar del conjunto factible dichas soluciones, se añade al Modelo SC un nuevo conjunto de restricciones, que crece de manera exponencial respecto del tamaño de una instancia del HPP. Lancia y Serafini proponen entonces un método por el cual las restricciones de este conjunto se van añadiendo a la formulación dinámicamente, según van siendo necesarias durante la resolución del problema. Sus resultados muestran que este modelo permite resolver problemas que no son resolubles mediante el Modelo RTIP.

El tercer y último de los modelos de optimización lineal entera es el propuesto por Catanzaro, Godi y Labbé en 2010. Al contrario que los anteriores, este es un modelo polinómico cuya idea se basa en el concepto de representante de clases, así como en algunas propiedades que cumplen los haplotipos de cualquier solución factible. Estudiaremos primero un Modelo Básico, y después veremos la forma de reducir el número de variables y restricciones de la formulación, dando lugar a un Modelo Reducido. Posteriormente, proponemos en este trabajo algunas variaciones del modelo que sirven para eliminar un mayor número de variables de decisión de la formulación, e incluimos en el mismo otros conjuntos de restricciones necesarios para que funcione correctamente. Estas variaciones no se incluyen en el artículo de Catanzaro et al., sino que nos han surgido a raíz del análisis del modelo y su implementación en Xpress. Ilustraremos la utilidad de las distintas restricciones, las diferencias entre los modelos y el proceso de eliminación de variables de decisión innecesarias a través de un ejemplo resuelto utilizando las formulaciones del Modelo Básico y del Modelo Reducido con las mejoras introducidas a posteriori, empleando para ello el programa de optimización Xpress. Catanzaro, Godi y Labbé proponen, además, conjuntos de desigualdades válidas que refuerzan el Modelo Reducido, y muestran que este modelo obtiene mejores resultados que los modelos de optimización lineal entera vistos hasta ahora y que cualquier otro anterior al mismo. Finalmente, incluimos en nuestro trabajo un estudio comparativo realizado con Xpress entre los modelos Básico, Reducido y Reducido con algunas desigualdades válidas adicionales para conjuntos con distinto número de SNP y de genotipos.

En el último capítulo, abordamos la resolución del problema HPP desde otra perspectiva, a través de un modelo teórico propuesto en 2010 por Matos Camacho, Schiermeyer y Tuza. A diferencia de los anteriores, este trabajo proporciona un algoritmo para obtener una solución aproximada del problema HPP, y se fundamenta en un problema de grafos, el Problema del Mínimo Subgrafo Arcoíris (PSAM). Este problema consiste en, dado un grafo cuyas aristas están coloreadas con p colores, encontrar un subgrafo del anterior que contenga p aristas, cada una de un color, y el menor número posible de vértices. En este capítulo comprobaremos que este problema guarda una estrecha relación con el problema HPP, y veremos un algoritmo que proporciona una solución aproximada en función del grado del grafo.

Para finalizar, se incluyen en el trabajo dos anexos. El anexo 1 contiene el código relativo a la formulación de los modelos Básico y Reducido, con las variaciones propuestas, que hemos elaborado e implementado en Xpress. El anexo 2 incorpora un algoritmo descrito por Matos Camacho et al. para obtener el máximo número de caminos de tres nodos arista-disjuntos dos a dos en un grafo conexo. Los pasos del algoritmo se ilustran en el anexo mediante un ejemplo que hemos diseñado para facilitar su comprensión.

Abstract

The Human Genome Project (HGP) was an international research which begun in 1990 with the purpose of sequencing the complete genome of our species, in other words, to decipher the DNA contained in each of the cells of a human being. Our DNA is organized in pairs of chromosomes, one of the copies is inherited from our father and the other one from our mother. It is possible to describe it by a sequence built out of the combination of four bases. More than 99 % of the base pairs within the sequence is the same for everybody. When the HGP ended in 2003, the sequence of more than three billions base pairs had already been deciphered, opening the doors to new investigations which are still being held and whose purpose is to establish the differences in the DNA among individuals. The differences between two people due to the alteration of exactly one of the basis of the sequence are known as SNP (from Single Nucleotide Polymorphism) and they are, together with the process of recombination, the main form of human genetic variation. On average, a SNP can be found in one chromosome every thousand basis. Knowing these differences is vital in the diagnosis and treatment of genetic disorders, as well as in the development of medicines and in drug-design. Furthermore, understanding it will allow us to widen our insights into evolution and will take us closer to the understanding of basic cellular processes, such as the protein synthesis and its function.

The processes by which the sequences of basis inherited from each one of the biological parents are obtained are very expensive and time-consuming. However, more accessible processes also allow us to obtain information about the chromosomes. Specifically, they provide information about the bases on each allele (which is a specific position of a chromosome), designating whether a given individual is *homozygous* (having identical pairs of genes for the allele) or *heterozygous* (having dissimilar base pairs for the allele). Nevertheless, in the case of an individual being heterozygous for an allele, they do not provide information on which basis has been contributed by each of the biological parents. This information is known as the *genotype* of an individual.

It is in this context that haplotyping problems emerged by the end of the 20th Century. Their aim is to haplotype an individual, in other words, to find out the sequence of basis contributed by each of the parents (their haplotype). The process of finding out the haplotype of a given population involves haplotyping each and every of the individuals, and can be carried out following different biological purposes. One of the most accepted criterion is that of the Pure Parsimony, which aims at finding out the minimum number of haplotipes that can lead to the formation of such population. One of the biological incentives that justify the use of this criterion is the fact that the number of haplotypes found in nature is much lower than the number of haplotypes we could deduct from the amount of SNP on human DNA, as it is subject to huge variation. What's more, if we proceed on the basis that we descend from a reduced number of ancesters, it seams reasonable to believe that their haplotypes are just the same as ours (excluding the cases of genetic recombination and mutations).

In this essay, we have studied different models proposed in the last two decades in order to tackle the Haplotyping problem by means of the criterion of Pure Parsimony.

The first chapter comprises the contents we will need to master in order to understand the following ones. The first section deals with the definition of several basic concepts of

Graph Theory which are considered necessary for the understanding of the theoretical model described in detail in the last chapter. A section with the definition of matching in a graph and some theorems and results used for finding out a maximum matching in a graph has also been included, as it will be useful for the last chapter. The definition of linear and integer optimization models has also been included at the end of this chapter, as they will prove themselves necessary in the third chapter.

The second chapter is an introduction to haplotyping problems. It deals with the origin of the problem, providing the biological incentives leading to its study. To this end, this chapter deals with the definition of every of the concepts frequently used for the explanation of the models, from a biological point of view. The second section of the chapter introduces a classification of haplotyping problems in two different groups, explaining their aims and some of the most common criterion used for solving them. This chapter, therefore, puts into context the Pure Parsimony Haplotyping (PPH) problem, which is the main object of this research.

The third chapter contains a detailed explanation of the presentation and formulations of four different integer optimization models proposed for approaching the PPH problem. The first section of the chapter comprises the mathematical notation used afterwards in the models description, as well as some definitions. The first model studied in this research is the TIP Model, proposed by Gusfield in 2003 as the first one to be formulated by integer linear optimization. This formulation involves the creation of a decision variable for each of the haplotypes that may have led to one of the genotypes of the population, and it aims at minimizing the number of haplotypes by trying to choose the maximum number of haplotypes involved in the creation of more than one genotype. Due to the fact that the number of haplotypes that may give rise to a genotype grows exponentially according to the number of its heterzygous alleles, the variables of the formulation also grow exponentially according to the size of the problem. Gusfield provides in the same article a means for lowering the high number of variables, leading to a new model known as the RTIP Model. However, several studies conducted prove that the RTIP model is not convenient for problems of size bigger than 30 SNP and 50 individuals.

The second model studied in this chapter is the Set Covering Model (SC). Proposed in 2009 by Lancia and Serafini, it is based on a set covering condition that is proved to be satisfied by the haplotypes of any feasible solution of the PPH problem. This condition states that, given a heterozygous position of a genotype, there must exist, in any of the feasible solutions, two haplotypes that might have generated this genotype, each one of them having in that position one of the two bases of the genotype. As this condition is necessary, but not sufficient, for any set of haplotypes to be optimal, this model constitutes a relaxation of the PPH problem. As it will be proved in this chapter, there are optimal solutions that meet this condition which are, however, unfeasible for the PPH problem. In order to remove these solutions from the set of feasible ones, a new set of restrictions will be added to the SC Model, despite it grows exponentially according to the size of an instance of the PPH. In order to deal with this drawback, Lancia and Serafini propose a method by which the restrictions of this set are dynamically added to the formulation right when they are needed during the process of solving the problem. The results prove that this model allows us to tackle problems for which the RTIP Model is not effective.

The third and last integer optimization model studied was proposed by Catanzaro,

Godi and Labbe in 2010. Unlike the former ones, this is a polynomial model, and its main idea is based on the concept of class representative, as well as on some of the properties met by the haplotypes of any feasible solution. First of all, a Basic Model will be introduced. Then, we shall see the way of lowering the number of decision variables and restrictions of the formulation, leading to the Reduced Model. Afterwards, we propose in our research some variations of the model which aim at removing a higher number of variables within the formulation, as well as establishing new sets of restrictions necessary for its correct functioning. These modifications were not proposed by Catanzaro et al., but rather have emerged through the analysis of the model and its implementation on Xpress. We will also exemplify the operation of the different restrictions, the differences between models and the processes for removing unnecessary decision variables by means of an example solved using the formulations of the Basic and the Reduced Model with the improvements previously introduced, along with the Xpress optimizer. Catanzaro, Godi and Labbé also include in their research several sets of valid inequalities to strengthen the Reduced Model, and they prove this model offers better results than all the previous ones. Finally, we include in this essay a study conducted using Xpress, comparing the Basic Model, the Reduced one and the latter one along with some of the additional valid inequalities. We use sets of different number of SNP and genotypes to conduct this study.

In the last chapter, we tackle the PPH problem from another point of view, by studying a theoretical model proposed in 2010 by Matos Camacho, Schiermeyer and Tuza. Unlike the previousy mentioned models, this research is supported by an algorithm which obtains an approximate solution for the PPH problem, and it is based on a graph problem, the Minimum Rainbow Subgraph problem (MSR). Given a graph whose edges are coloured with p colours, this problem aims at finding the minimum subgraph with p edges, each one of a different colour, and the minimum number of vertices. In this chapter, we will establish the relationship between this problem and the PPH one, and an algorithm to find an approximate solution according to the graph degree. To sum up, two appendices are included in this research. The first appendix contains the code of the Basic and the Reduced Model formulations, including the proposed changes, introduced in Xpress. The second appendix introduces an algorithm described by Matos Camacho et al. in order to obtain the maximum number of pairwise edge disjoint paths of order three in a connected graph. The steps of the algorithm are ilustrated in this research by means of an example we have supported to enable its understanding to the reader .

Capítulo 1

Conceptos previos

En este capítulo, vamos a incluir distintos conceptos que emplearemos a lo largo del trabajo. En la primera sección, vamos a centrarnos en algunas definiciones y conceptos básicos en Teoría de Grafos, que han sido estudiados durante el Grado en Matemáticas en la asignatura *Grafos y Optimización Discreta*, y se pueden encontrar en [6]. En la segunda sección, nos ocuparemos de definir un emparejamiento en un grafo y de dar solución al problema de encontrar un emparejamiento máximo para un grafo G, viendo primero algunas definiciones y conceptos necesarios para ello. Los contenidos de esta sección no se estudian en el grado, por lo que se pueden encontrar en [8, pp. 209-220]. Por último, en la tercera sección veremos los conceptos de problema de optimización lineal y problema de optimización lineal entera, conceptos estudiados en el grado y que se pueden encontrar en [6].

1.1. Breve introducción a la Teoría de Grafos

Definición 1.1. Un *grafo no dirigido* es un par (V, E) formado por un conjunto finito $V \neq \emptyset$, a cuyos elementos denominaremos *vértices o nodos* y E, un conjunto de pares no ordenados de elementos de V a los que llamaremos *aristas*. Un *bucle* es una arista de la forma (v_i, v_i). Un grafo sin bucles se llama *simple*.

- A los nodos v_1, v_2 se les llama *extremos* de la arista e si $e = (v_1, v_2)$, y en tal caso se dice que los nodos son *adyacentes o vecinos*.
- Una arista e es *incidente* con un vértice v si v es uno de sus extremos. En ese caso, se dice que e *incide* en v.

Representaremos como $V(G)$ (o V) al conjunto de nodos de un grafo G, y como $E(G)$ (o E) al conjunto de aristas de G.

Un grafo se puede representar mediante un diagrama en el que cada vértice está asociado a un objeto (normalmente un círculo o una circunferencia), y cada arista se representa mediante una línea que une los dos vértices que la componen.

Definición 1.2. Se llama *orden* de un grafo a $|V|$ (se suele denotar por n). Se llama *tamaño* de un grafo a $|E|$ (se suele denotar por m).

Definición 1.3. El *grado de incidencia* $g(v)$ de un vértice v es el número de aristas que inciden en él. El *grado máximo* de un grafo es el máximo de los grados de todos sus vértices.

Definición 1.4. Dado un grafo $G = (V, E)$, se dice que otro grafo $G' = (V', E')$ es *subgrafo* de G si $V' \subseteq V$ y $E' \subseteq E$. Dado un grafo $G = (V, E)$ y dado $B \subseteq V$, se llama *subgrafo de G inducido* por B a $G_B = (B, E_B)$, donde

$$E_B = \{(i, j) \in E : i \in B, j \in B\}.$$

Definición 1.5. Se llama grafo *completo* a $G = (V, E)$ tal que $v_1, v_2 \in V$, $v_1 \neq v_2 \Rightarrow (v_1, v_2) \in E$.

Definición 1.6. Un grafo $G = (V, E)$ es *bipartito* si $V = V_1 \cup V_2$ con $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V_1 \cap V_2 = \emptyset$ y

$$\forall (v_1, v_2) \in E \Rightarrow \begin{cases} v_1 \in V_1, & v_2 \in V_2 \qquad \text{o bien} \\ v_1 \in V_2, & v_2 \in V_1. \end{cases}$$

Definición 1.7. Llamamos *paseo* en un grafo $G = (V, E)$ a una sucesión alternante de aristas de E y vértices de V, $(v_1, (v_1, v_2), v_2, (v_2, v_3), v_3, \ \ldots\ , v_{k-1}, (v_{k-1}, v_k), v_k)$, que comienza en un vértice y termina en otro, tal que cada arista incide en el nodo anterior y posterior de la sucesión. Si $v_1 = v_k$, se dice que el paseo es *cerrado*.

Definición 1.8. Una *cadena o camino* es un grafo $P = (V, E)$ definido por $V = \{v_1, \ldots, v_n\}$ y $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$, es decir, una sucesión alternante de vértices y aristas distintos que comienza en un vértice y termina en otro, tal que cada arista incide en el nodo anterior y posterior de la sucesión. Se dice que P *conecta* v_1 con v_n. Un camino con m aristas se dice de *longitud* m y lo denominaremos P_{m+1}.

Definición 1.9. Un *ciclo* (o ciclo elemental) es un grafo $C = (V, E)$ definido por $V = \{v_1, \ldots, v_n\}$, con $n \geq 3$, y $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}$, es decir, una cadena más una arista entre los vértices unidos por la cadena. Un subgrafo de G que sea un ciclo se denominará ciclo en el grafo G.

Definición 1.10. Un grafo es *conexo* si cualquier par de vértices (distintos) están conectados por una cadena. Un grafo que no es conexo suele llamarse *disconexo*.

Definiendo en V la relación siguiente:

$$\forall v_1, v_2 \in V, \qquad i \ \mathsf{C}\ j \Leftrightarrow \begin{cases} v_1 = v_2 & \text{o bien} \\ \\ \text{existe una cadena en } G \text{ que conecta } v_1 \text{ con } v_2. \end{cases}$$

C es una relación de equivalencia, de forma que $V|\mathsf{C}$ constituye una partición de V, cada una de cuyas clases de equivalencia se denomina *componente conexa*.

Dado $G = (V, E)$ conexo, si $(V, E \backslash \{e\})$ es disconexo, la arista e se denomina *puente*.

Definición 1.11. Diremos que un grafo $G = (V, E)$ es un *árbol* si es conexo y no contiene ciclos. Un *árbol generador* de G es un subgrafo $G = (V, E')$, con $E' \subseteq E$, conexo y sin ciclos. En un árbol, los nodos con grado de incidencia 1 se denominan *hojas*.

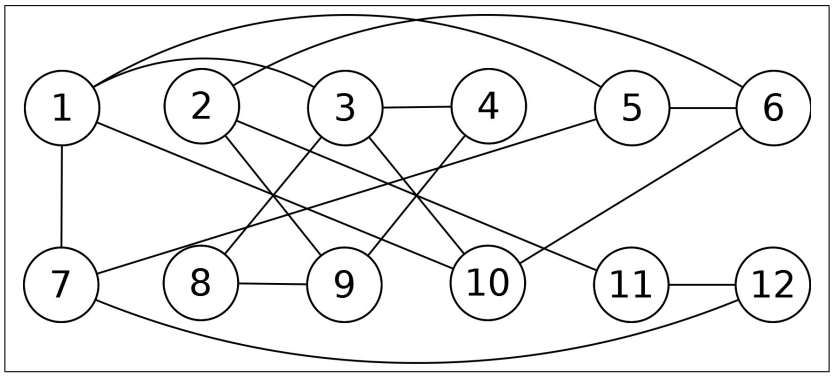


Figura 1.1: Grafo G.

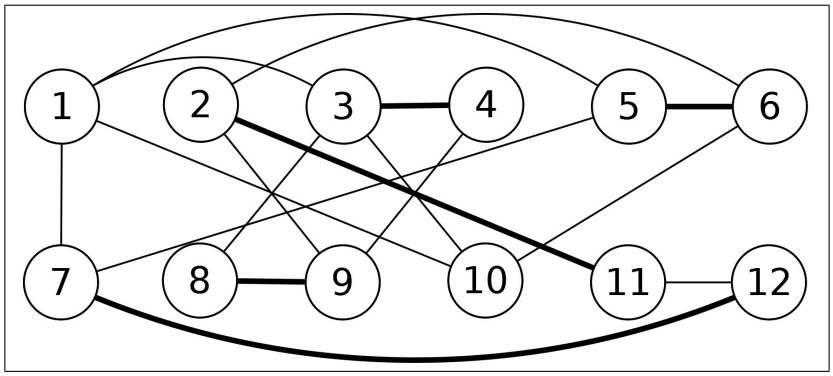


Figura 1.2: Emparejamiento Q en G.

1.2. Emparejamiento máximo

Definición 1.12. Dado un grafo no dirigido $G = (V, E)$, decimos que un subconjunto Q de aristas es un *emparejamiento* si dos aristas de Q no inciden sobre un mismo vértice. Si el emparejamiento tiene cardinalidad máxima, decimos que es un *emparejamiento máximo*.

En la Figura 1.1 se puede observar que el conjunto $Q_1 = \{(1,5),(3,8),(5,6)\}$ no es emparejamiento, $Q_2 = \{(1,3),(2,6),(5,7)\}$ es emparejamiento pero no es máximo, y $Q_3 = \{(1,10),(2,11),(3,4),(5,6),(7,12),(8,9)\}$, $Q_4 = \{(1,5),(2,11),(3,8),(4,9),(6,10),(7,12)\}$ son emparejamientos máximos. Es claro que un emparejamiento tiene a lo sumo $\lfloor \frac{|V(G)|}{2} \rfloor$ aristas.

Definición 1.13. Dado un emparejamiento Q, llamamos *vértice emparejado* a todo vértice i en el cual incida una arista de Q; en caso contrario, decimos que i es un *vértice libre*.

Es evidente a partir de la definición que si solo existe un vértice libre respecto de un emparejamiento Q, entonces Q es máximo.

Definición 1.14. Llamamos *cadena alternante* a toda cadena L tal que para cualquier par de aristas consecutivas $e, e' \in L$ tenemos $e \in Q, e' \notin Q$, o bien $e \notin Q, e' \in Q$. Una *cadena de aumento* es una cadena alternante que comienza en un vértice libre i_0 y finaliza en un vértice libre $j_0 \neq i_0$. Todo vértice de orden par en una cadena alternante se conoce con el nombre de *vértice externo*, y si es de orden impar, con el nombre de *vértice interno*

(suponiendo que el vértice inicial es de orden 0, y el resto de vértices v_i tienen orden dado por la longitud del camino que conecta la raíz a v_i).

En la Figura 1.2 podemos ver un emparejamiento del grafo de la Figura 1.1. Para este emparejamiento, la cadena $L = \{(1,5),(5,6),(6,10)\}$ es una cadena de aumento entre los vértices 1 y 10 en la que los vértices $5, 10$ son internos y los vértices $1, 6$ son externos.

Se tiene ahora un resultado de caracterización de los emparejamientos máximos en términos de cadenas de aumento:

Teorema 1.1. *Un emparejamiento Q es máximo sii no existe ninguna cadena de aumento respecto de Q.*

Demostración.
$\Longrightarrow$ Supongamos que $\exists L$ cadena de aumento respecto de un emparejamiento Q. Denotando por $Q - L$ al conjunto de aristas que pertenecen a Q y no pertenecen a L (y viceversa), se tiene $Q^{'} = (Q - L) \cup (L - Q) =: Q\Delta L$ es también emparejamiento y $|Q^{'}| = |Q| + 1$. Por tanto, Q no es máximo.

$\Longleftarrow$ Recíprocamente, supongamos que no existe ninguna cadena de aumento respecto de Q y sea $|Q^{'}|$ tal que $|Q^{'}| > |Q|$. Si consideramos el grafo $G^{'} = (V, Q\Delta Q^{'})$, en cada vértice de V inciden a lo más dos aristas, una de Q y otra de $Q^{'}$. Por consiguiente, las componentes conexas de $G^{'}$ son vértices aislados o cadenas formadas por aristas de Q y $Q^{'}$ de una forma alternante. Como $|Q^{'}| > |Q|$, tiene que existir una de dichas cadenas L que comience y termine con aristas de $Q^{'}$, pero entonces L sería de aumento respecto de Q, lo cual es una contradicción. Por lo tanto, Q es máximo. □

Tras lo visto anteriormente, es claro que si L es cadena de aumento respecto de Q, entonces $Q^{'} = Q\Delta L$ es un emparejamiento tal que $|Q^{'}| = |Q| + 1$.

1.2.1. Árboles alternantes

En este apartado vamos a introducir el concepto de árbol alternante, explicar su construcción y características y su utilidad en la construcción de emparejamientos máximos.

Sea $G = (V, E)$ un grafo y Q un emparejamiento de G. Como hemos visto en el apartado anterior, que Q sea máximo equivale a que no existan cadenas de aumento L con respecto de Q. Por tanto, un método para determinar si Q es máximo es tomar un vértice libre r en V y construir sucesivamente cadenas alternantes que partan de r hasta encontrar una cadena de aumento. Si encontramos una, entonces Q no es máximo, y si de r no parte ninguna cadena de aumento, pasamos a examinar otro vértice libre. Este proceso se repite hasta que se encuentra una cadena de aumento o hasta que se concluye que no existen cadenas de aumento, en cuyo caso Q es máximo.

Podemos construir el conjunto de cadenas alternantes que parten de un vértice libre r mediante la creación de un árbol que incluirá al nodo r, al que llamaremos raíz, y en el que los vértices que están a nivel impar son vértices internos de una cadena alternante y los que están a nivel par son vértices externos (r se encuentra en el nivel 0). Este árbol se conoce con el nombre de *árbol alternante* y en él pueden aparecer vértices de G

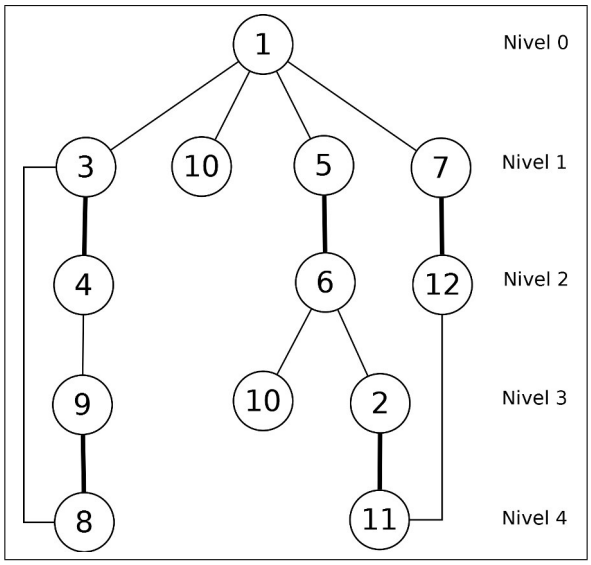


Figura 1.3: Árbol alternante con raíz $r = 1$ construido a partir del emparejamiento Q.

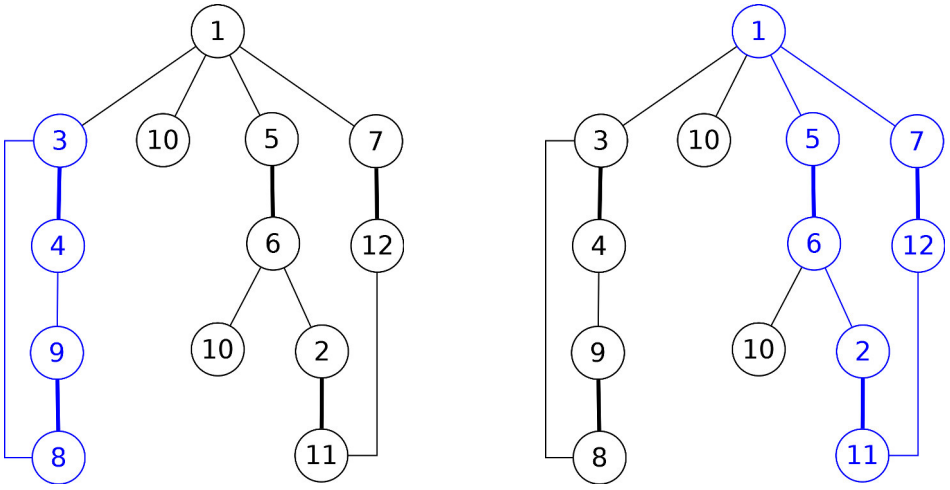
repetidos, pero únicamente en los niveles impares (vértices internos). Por tanto, es claro que para cada vértice externo i del árbol existe una única cadena alternante de r a i. En la Figura 1.3 podemos ver un árbol alternante con raíz el vértice $r = 1$ asociado al emparejamiento de la Figura 1.2. En este árbol podemos apreciar que las cadenas $L_1 = \{(1, 10)\}$ y $L_2 = \{(1, 5), (5, 6), (6, 10)\}$ son las únicas cadenas de aumento partiendo de esta raíz.

Si en el proceso de construcción de un árbol alternante encontramos una cadena de aumento del nodo raíz a un nodo i libre, i estará en un nivel impar del árbol. En este caso, haciendo $Q' = Q\Delta L$ obtenemos un emparejamiento Q' donde r, i no son libres. Si no quedan vértices libres, Q' es de aumento, y si quedan, repetimos el proceso tal y como se ha explicado anteriormente hasta encontrar un emparejamiento máximo.

Como un vértice interno es adyacente a otro externo del siguiente nivel únicamente mediante una arista del emparejamiento, y los vértices externos no se repiten, en un árbol alternante no se dan ciclos formados mediante aristas de G que conecten un vértice interno con otro del mismo tipo. No obstante, sí pueden existir aristas cuyos extremos sean un vértice externo y otro interno, o bien dos vértices externos:

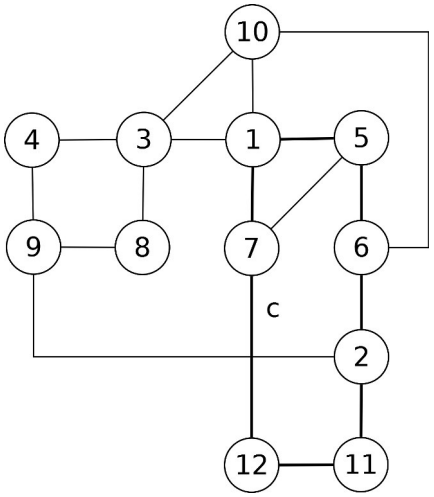
- Si una arista es extremo de un vértice interno y otro externo, da lugar a un ciclo con un número par de aristas, la mitad de las cuales están en Q. Estos ciclos se pueden ignorar, ya que a través de ellos no se pueden formar cadenas de aumento. Podemos ver un ejemplo en la Figura 1.4a.

- Si una arista tiene por extremos dos vértices externos, da lugar a un ciclo de tamaño impar, a partir del cual sí es posible obtener cadenas de aumento en el grafo. En la Figura 1.4b podemos ver un ejemplo de ciclo impar. Este tipo de ciclos se estudian en el apartado siguiente.

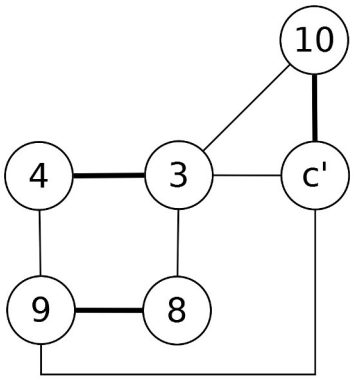


(a) Ciclo par. (b) Ciclo impar.

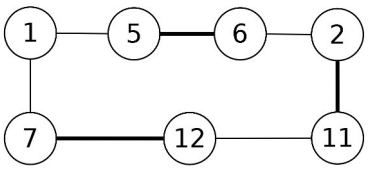
Figura 1.4: Ejemplos de ciclos par e impar en un árbol alternante.



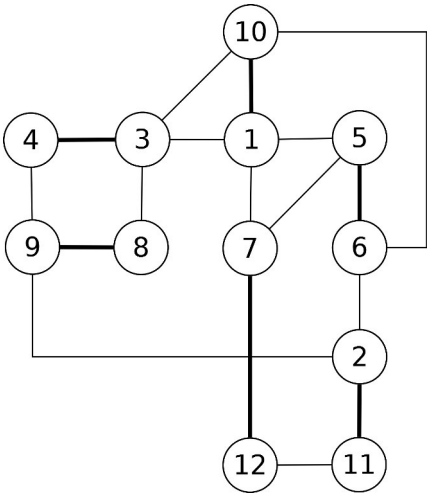
(a) Grafo G y ciclo c.



(b) Grafo G/c y emparejamiento Q_1.



(c) Emparejamiento Q_c en c.



(d) Emparejamiento Q en G.

Figura 1.5: Construcción de Q a partir de Q_1.

Grafos contraídos y blossoms

Definición 1.15. Dado μ un ciclo elemental con un número impar de aristas, llamamos *grafo contraído de G por μ* al grafo $G/\mu = (V/\mu, E/\mu)$, donde $V/\mu = (V - V_\mu) \cup \{v_\mu\}$, siendo V_μ el conjunto de vértices de μ y v_μ un nuevo vértice (*contracción de μ*), y $E/\mu = \{(i,j) : (i,j) \in E,\ i,j \in V - V_\mu\} \cup \{(i,v_\mu) : (i,j) \in E \text{ para algún } j \in V_\mu\}$. Es decir, un grafo que incluye todos los vértices de G que no pertenecen al ciclo μ y todas las aristas que inciden sobre dos de estos vértices, y además incluye un vértice nuevo v_μ y las aristas de la forma (v_i, v_μ) solamente si existía una arista que era extremo del vértice v_i y algún vértice del ciclo μ.

Se verifica la siguiente:

Proposición 1.1. *Si Q_1 es un emparejamiento en G/μ, entonces $\exists Q_\mu$ emparejamiento máximo en μ tal que el conjunto de aristas que pertenecen a Q_1, Q_μ ($Q = Q_1 \cup Q_\mu$) es un emparejamiento en G.*

Demostración. Si Q_1 no tiene ninguna arista incidente en v_μ, entonces para cualquier emparejamiento máximo Q_μ en μ se verifica que $Q = Q_1 \cup Q_\mu$ es un emparejamiento. Supongamos ahora que (i, v_μ) es la única arista de Q_1 en G/μ que incide en v_μ, y sea j_0 un vértice de V_μ tal que $(i, j_0) \in E$ (esta arista existe por la definición de E/μ). Entonces, si Q_μ es el emparejamiento máximo en μ que deja libre j_0, se verifica que $Q = Q_1 \cup Q_\mu$ es un emparejamiento en G. □

El proceso de construcción de dicho emparejamiento Q en G a partir del emparejamiento Q_1 en G/μ se puede observar en la Figura 1.5.

Definición 1.16. Dado un emparejamiento Q, llamamos *blossom* a cualquier ciclo elemental μ con $2k+1$ aristas, de las cuales $2k$ pertenecen a Q.

En la Figura 1.5(c) tenemos un ejemplo de blossom. Además, es claro que si μ es un blossom, entonces el emparejamiento Q restringido a μ, $Q_\mu = Q \cap \mu$, es máximo respecto de μ. Veamos ahora un teorema necesario para determinar las cadenas de aumento en un árbol con un blossom:

Teorema 1.2. *Supongamos que Q es un emparejamiento con al menos dos vértices libres, y que en el proceso de construcción de un árbol alternante con raíz i_0 encontramos un blossom μ. Sea j_0 el vértice libre de μ respecto de q_μ tal que i_0 está unido a j_0 por una cadena alternante par L_0. Entonces:*

$$Q \text{ es máximo en } G \iff Q_1 = Q - Q_\mu \text{ es máximo en } G/\mu.$$

Demostración.
$\Longrightarrow$ Dado Q máximo en G, Q_1 es máximo en G/μ, ya que si $|Q_1'| > |Q_1|$ para algún emparejamiento $|Q_1'|$ en Q/μ, entonces por la Proposición 1.1 existe Q_μ' máximo en μ tal que $Q' = Q_\mu'$ es emparejamiento, luego se verificaría $|Q'| = |Q_1'| + |Q_\mu'| = |Q_1'| + |Q_\mu| > |Q_1| + |Q_\mu| = |Q|$, lo cual es absurdo.

$\Longleftarrow$ Recíprocamente, supongamos que Q_1 es máximo. Sean $Q^{'} = Q\Delta L_0$ y $Q_1^{'} = Q_1\Delta L_0$. Respecto de $Q^{'}$, i_0 estará emparejado y j_0 será libre, y respecto de $Q_1^{'}$, v_μ será libre. Como L_0 es par, $|Q_1^{'}| = |Q|$ y por tanto $Q_1^{'}$ es también máximo en Q/μ. Si $Q^{'}$ no fuera máximo respecto de G, existiría una cadena de aumento L respecto de $Q^{'}$. L debe incidir en μ, pues de lo contrario L sería de aumento respecto de $Q_1^{'}$ en G/μ, lo cual es imposible pues $Q_1^{'}$ es máximo en G/μ. Sea $i_1 \neq j_0$ un vértice libre de L en μ. La arista de L_{i_1} que incide en μ no puede pertenecer a $Q^{'}$, pues j_0 es libre respecto de $Q^{'}$ y los otros vértices de μ están emparejados con aristas de Q_μ. Entonces i_1 y v_μ serían vértices libres respecto de $Q_1^{'}$ y estarían unidos por la cadena alternante L_{i_1}, lo cual es absurdo pues $Q_1^{'}$ es máximo respecto de G/μ. En consecuencia, $Q^{'}$ es máximo respecto de G. Como $|Q^{'}| = |Q|$ pues L es par, resulta que Q es máximo respecto de G. □

1.2.2. Determinación de un emparejamiento máximo

Utilizando los resultados anteriores, en este apartado vamos a construir un algoritmo que determine un emparejamiento máximo Q en un grafo G a partir de un emparejamiento inicial construido por cualquier procedimiento. El siguiente algoritmo permite detectar un emparejamiento máximo basándose en la construcción de árboles alternantes:

Algoritmo 1.1.

1. *Hacer $Q = Q_0$, siendo Q_0 cualquier emparejamiento.*
2. *Si Q deja a lo más un vértice libre, Q es máximo. Parar.*

   Si no, ir a 3.
3. *Construir un árbol alternante cuya raíz sea un vértice libre respecto de Q.*
4. *Si el proceso de construcción termina detectando una cadena de aumento L, determinar $Q^{'} = Q\Delta L$ en el grafo $Q^{'}$, reconstruir el correspondiente emparejamiento en el grafo G e ir a 2.*

   Si el procedimiento termina con un árbol alternante completo y existen más vértices libres, ir a 3 con el subgrafo inducido por los vértices no contenidos en ninguno de los árboles alternantes completos previamente examinados.

   En caso contrario, parar. Q es máximo.

Como emparejamiento inicial Q_0 puede empezarse con un *emparejamiento completo*, es decir, que no contenga vértices libres adyacentes. Una forma de construirlo es emparejar sucesivamente vértices adyacentes (que no hayan sido emparejados previamente), eligiendo en cada momento los de menor grado de incidencia.

El proceso de construcción de un árbol alternante, Paso 3 del Algoritmo 1.1, se explica a continuación mediante el Algoritmo 1.2. Dado un vértice libre i_0 en Q, queremos construir un árbol alternante $T_a = (V_a, E_a)$, para el que $V_{ex} \subset V_a$ denotará los vértices externos y L_i la cadena que une i_0 con i en T_a.

Algoritmo 1.2. *Hacer $V_a = V_{ex} = \{i_0\}$, $E_a = \emptyset$, $G^{'} = G$.*

En cada iteración, se presenta solo uno de los siguientes casos:

1. *Detección de una cadena de aumento.*

   Si para algún $i \in V_{ex}$ existe $j \notin V_a$, $(i,j) \in E$ y j libre, hacer $L = L_i \cup (i,j)$ y parar.

2. *Aumentar el tamaño de T_a.*

   Si para algún $i \in V_{ex}$ existe $j \notin V_a$, $(i,j) \in E$ y j emparejado con algún vértice k, hacer $V_a = V_a \cup \{j\} \cup \{k\}$, $E_a = E_a \cup (i,j) \cup (j,k)$, $V_{ex} = V_{ex} \cup \{k\}$.

3. *Detección de un blossom.*

   Si para algún $i \in V_{ex}$ existe una arista (i,j) tal que $j \in V_{ex}$, hacer $\mu = (L_i \Delta L_j) \cup (i,j)$ (μ es un blossom), $G' = G'/\mu$, $T_a = T_a/\mu$.

4. *Detección de un árbol alternante completo.*

   Si no se da ninguno de los casos anteriores, el árbol alternante se dice completo. Parar.

En el cuarto capítulo del trabajo, emplearemos el algoritmo de determinación de un emparejamiento máximo como parte de otro algoritmo que nos servirá para construir un subgrafo arcoíris, sobre el que basaremos los teoremas y resultados posteriores.

1.3. Modelos de Optimización Lineal y Entera

Definición 1.17. Un *problema de optimización lineal* es aquél que se puede establecer mediante la formulación

$$\begin{cases} \text{mín}_x & c \cdot x \\ \text{s.a} & Ax \leq b \\ & x \in \mathbb{R}^n_+ \end{cases}$$

con $c \in \mathbb{R}^n$, $A_{mxn} = (a_{ij})$, $a_{ij} \in \mathbb{R}$ $\forall i, j, b \in \mathbb{R}^m$.

Es decir,

$$\begin{cases} \text{mín} & c_1x_1 + c_2x_2 + ... + c_nx_n \\ \text{s.a} & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leq b_2 \\ & \cdot_. \\ & a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \leq b_m \\ & x_1, x_2, ..., x_n \geq 0. \end{cases}$$

Definición 1.18. Las variables $x = (x_1, \ ... \ , x_n)$ se denominan *variables de decisión*. Las desigualdades $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ se denominan *restricciones lineales*. Las desigualdades $x_j \geq 0$ se denominan *restricciones de no negatividad*. La función lineal $\sum_{j=1}^{n} c_j x_j$ se denomina *función objetivo*. El conjunto de soluciones (valores de las variables) que satisfacen todas las restricciones se denomina *región factible*.

Definición 1.19. El valor mínimo que se busca determinar recibe el nombre de *valor óptimo* del problema. Puede no existir si:

- No se puede dar valores a las variables que satisfagan todas las restricciones, en cuyo caso se dice que el problema es *infactible*.
- Para cualquier $h \in \mathbb{R}$ existe un punto en la región factible x con $cx < h$. En este caso, se dice que el problema es *no acotado*.

Las soluciones que corresponden al valor óptimo se llaman *soluciones óptimas*.

Definición 1.20. Un *problema de optimización (lineal) entera* es aquel que se puede establecer mediante la formulación

$$\begin{cases} \text{(P) mín} & c \cdot x \\ \text{s.a} & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{cases}$$

con $c \in \mathbb{R}^n$, $A_{mxn} = (a_{ij})$, $a_{ij} \in \mathbb{R}\ \forall i, j, b \in \mathbb{R}^m$.

Es decir, añadiendo restricciones de integridad $x_j \in Z$ sobre las variables de un problema lineal.

Nótese que, en este caso, las restricciones lineales definen un poliedro dentro del cual se encuentran las soluciones factibles del problema (aquéllas que toman valores enteros). Por esta razón, usando las mismas variables se pueden construir infinidad de problemas con distintos tipos de restricciones pero con la misma región factible.

Definición 1.21. Llamamos *relajación lineal* de (P), (LP), al problema de optimización lineal que se obtiene suprimiendo las restricciones de integridad. Denotaremos con $v(P)$ al valor óptimo de un problema (P).

Se satisface la siguiente relación entre el valor óptimo de un problema (de minimización) y el de su relajación lineal:

$$v(LP) \leq v(P).$$

Definición 1.22. Se llama *salto de dualidad* a la diferencia $v(P) - v(LP)$.

Capítulo 2

Los problemas de haplotipaje

2.1. Introducción a los problemas de haplotipaje y motivación

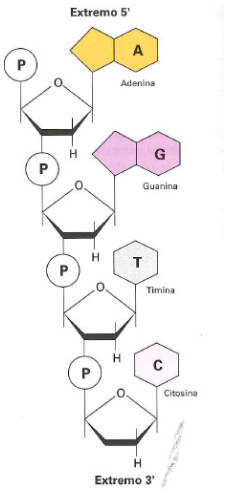
Para poder comprender la formulación de los problemas de haplotipaje, necesitamos primero introducir algunos conceptos de Genética, en particular los relacionados con la estructura del ADN, el Proyecto Genoma Humano (PGH) y las diferencias en el genoma entre distintas personas. Estos contenidos se pueden encontrar en [10, pp. 577-594], [9, pp. 552-556], [3, pp. 128-132, 190] y [4].

2.1.1. Estructura del ADN

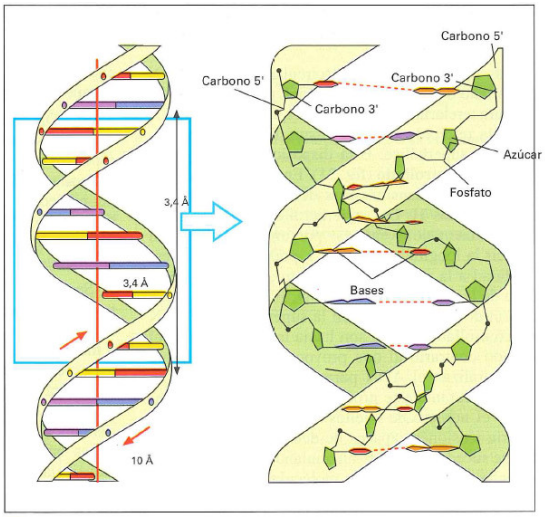
El ADN es una macromolécula formada por unidades similares, que llamaremos nucleótidos, enlazadas por sus extremos dando lugar a cadenas. Cada uno de estos nucleótidos consta de tres componentes: un azúcar, un fosfato y una base. El azúcar es siempre la desoxirribosa, y junto con el fosfato da lugar a lo que se conoce como el esqueleto del ADN, que es constante en cada nucleótido. Sin embargo, las *bases* varían de un nucleótido a otro, y es la secuencia de bases la que caracteriza de forma exclusiva al ADN y la que contiene la información de los genes y el material hereditario. Existen cuatro bases en el ADN, llamadas adenina (A), timina (T), citosina (C) y guanina (G). En la Figura 2.1a podemos ver un esquema de la estructura primaria del ADN.

En los organismos superiores, como por ejemplo los humanos, la estructura de las cadenas o hebras de ADN permite su disposición en forma de hélice doble, de manera que las dos hebras (dispuestas en direcciones opuestas) se enrollan en torno a un eje, dejando los esqueletos azúcar-fosfato en la parte externa y las bases en el interior. Las bases se emparejan de forma que la A siempre va con la T, y la C siempre con la G. Así, si se conoce la secuencia de una de las hebras, la secuencia de la otra queda perfectamente definida, hecho clave en la replicación del material genético. Esta estructura secundaria podemos observarla en la Figura 2.1b, que hemos tomado de [3].

Las dobles hélices del ADN puede adoptar múltiples formas. En concreto, para disminuir su longitud y así poder situarse dentro del núcleo de una célula humana, estas dobles



(a) Esquema de la secuencia de nucleótidos del ADN.



(b) Estructura secundaria del ADN: la doble hélice.

Figura 2.1: Estructuras primaria y secundaria del ADN.

hélices tienen la capacidad de enrollarse sobre sí mismas, dando lugar a lo que constituye una estructura terciaria o superenrollado.

Una molécula de ADN tiene que estar formada por muchos nucleótidos para albergar la información genética necesaria para incluso el más sencillo de los organismos. Concretamente, el genoma humano consta de aproximadamente 3000 millones de pares de bases distribuidas en 24 moléculas distintas de ADN, los cromosomas (22 autosomas más dos cromosomas sexuales). Así, el ADN de cada individuo se compone de 23 pares de cromosomas, y para cada par, uno de los cromosomas se hereda del padre y el otro de la madre.

2.1.2. Diferencias en el ADN entre los seres humanos

El Proyecto Genoma Humano (PGH) se inició en 1990 con los objetivos de determinar la secuencia de nucleótidos del genoma humano, identificar los genes y describir sus funciones. La secuencia del ADN logró determinarse por completo en la primavera de 2003, y ha supuesto un enorme beneficio en el campo de la Medicina, por ejemplo para el diagnóstico de enfermedades y el desarrollo de tratamientos, y en otros campos como el de la Biotecnología. A raíz de completarse la secuencia, los científicos se han volcado en encontrar las diferencias en las secuencias de nucleótidos entre cada persona, proceso que sigue en constante investigación.

Los estudios en la variación del genoma humano han determinado que las diferencias visibles entre dos individuos de nuestra especie se deben a variaciones en su secuencia de ADN que suponen menos del 0.1 %, lo que en la práctica supone una diferencia en más de 3 millones de pares de bases entre un individuo y otro. Cuando la posición de un nucleótido específico muestra una variabilidad estadísticamente significativa en una

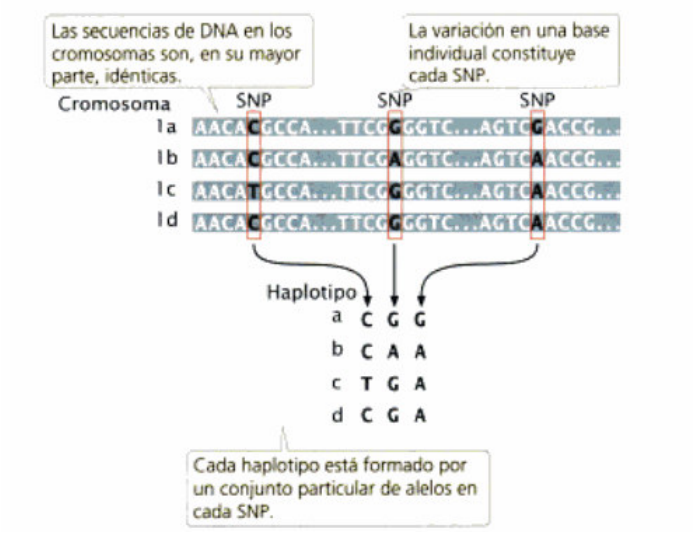


Figura 2.2: Obtención de haplotipos a partir de los SNP observados en una cadena de un mismo cromosoma en cuatro individuos distintos.

población, se le llama *polimorfismo de nucleótido único o SNP* (del inglés Single Nucleotide Polymorphism, pronunciado "snip"). Concretamente, una posición se considera un SNP si para una minoría de la población se observa un nucleótico (el llamado *alelo* menos frecuente), mientras que para el resto de la población se observa un nucleótido distinto (el alelo más frecuente). Por tanto, un SNP es casi siempre un polimorfismo con sólo dos alelos (de los cuatro alelos posibles dados por las cuatro bases A, T, C y G). Dado un SNP, un individuo puede ser *homocigótico* (es decir, poseer el mismo alelo en los dos cromosomas) o *heterocigótico* (es decir, poseer alelos distintos). Los polimorfismos de un nucleótido único son muy frecuentes y están presentes a lo largo de todo genoma. Si comparamos un mismo cromosoma en dos individuos, podremos hallar un SNP de media cada 1000 nucleótidos. Los SNP son la forma predominante de variación genética humana (excluyendo el proceso de recombinación), y su estudio tiene aplicación en Medicina, diseño de drogas, diagnóstico y en el campo forense, entre otros.

El grupo específico de SNP (y de otras variantes genéticas) que se observa en un cromosoma constituye un *haplotipo*. En la Figura 2.2, que podemos encontrar en [9], tenemos un ejemplo de la formación de haplotipos a partir de los SNP encontrados en una cadena de los cromosomas 1a, 1b, 1c y 1d, que representan copias diferentes de cadenas que podrían darse en individuos de una población. En este ejemplo, los alelos en el SNP 1 son T y C, mientras que los alelos de los SNP 2 y 3 son A y G.

Haplotipar un individuo consiste en determinar los dos haplotipos que componen un cromosoma, y es un proceso que, hecho de forma experimental, puede ser difícil y/o muy costoso de realizar, además de requerir en algunos casos gran cantidad de tiempo.

Los problemas de haplotipaje surgen para obtener los haplotipos o corregirlos en el caso de errores cometidos de forma experimental. Estos problemas han captado una gran atención estos últimos años debido a su importancia en el análisis de datos genéticos a pequeña escala. Por ejemplo, los haplotipos son necesarios en estudios de evolución para extraer la información necesaria para detectar enfermedades y reducir el número de análisis a llevar a cabo; en Genómica Funcional, los haplotipos se utilizan para descubrir

interacción entre genes o estudiar una respuesta alterada de un organismo a un tratamiento particular; en Farmacogenética, el estudio de haplotipos de una población sirve para explicar por qué las personas reaccionan de forma distinta a diferentes tipos o cantidades de medicamentos. Los métodos de haplotipaje vía simulación computacional resultan alternativas atractivas y en algunos casos, constituyen el único procedimiento viable de haplotipar poblaciones. Hay dos tipos de problemas de haplotipaje, que estudiaremos en el siguiente apartado.

2.2. Tipos de problemas de haplotipaje

Basándonos en [4], podemos establecer dos tipos de problemas de haplotipaje diferenciados: los problemas de haplotipaje de un individuo, y los problemas de haplotipaje de un conjunto de individuos o población.

2.2.1. Haplotipaje de Individuos

Como hemos mencionado anteriormente, el proceso de obtención de una cadena de bases nitrogenadas (una cadena de Aes, Tes, Ces y Ges que representan, respectivamente, la adenina, la timina, la citosina y la guanina) a partir de una secuencia de nucleótidos de una molécula de ADN es conocido como *secuenciación*, y su mayor problema es que, debido a limitaciones tecnológicas, hoy en día no es factible secuenciar una cadena larga de ADN de una vez. Una de las técnicas más utilizadas en el Proyecto Genoma Humano para la secuenciación del ADN, y que aún se utiliza para la determinación de las variaciones en el genoma entre individuos, es la conocida como *secuenciación del genoma completo por fragmentos escogidos al azar* (o shotgun, del inglés).

De manera resumida, este proceso consiste en secuenciar fragmentos cortos de ADN (llamados lecturas), de entre 300 y 1000 nucleótidos de longitud aproximadamente en una sola reacción, y ensamblarlos después. Por tanto, si queremos secuenciar una molécula completa de ADN, es necesario replicarla primero, creando numerosas copias, en un proceso que se conoce como *amplificación*. Entonces estas copias se rompen en fragmentos más pequeños, y aquellos que tengan un tamaño apropiado serán secuenciados. Las copias nos servirán para que, durante el proceso, no se deje ningún intervalo de nucleótidos de la molécula sin secuenciar. Posteriormente, mediante un proceso de *ensamblaje*, se van superponiendo las lecturas que se solapan entre ellas utilizando algoritmos, hasta que se determina la secuencia de ADN original. Los pasos de este proceso se ilustran en la Figura 2.3.

La mayor dificultad que se encuentra durante el proceso de ensamblaje de secuencias es que, en la fase anterior de amplificación, se replican juntas las copias de los cromosomas materno y paterno, pero se desconoce qué fragmentos pertenecen a cada progenitor (salvo por algunos fragmentos del final de la cadena de nucleótidos). Además, incluso con la mejor tecnología disponible, son inevitables errores en la secuenciación. Algunos son debidos a la presencia de contaminantes, como ADN que no pertenezca a la muestra a secuenciar, y otros consisten en la identificación errónea de bases nitrogenadas o en su falta de identificación. Debido a estos errores experimentales, el proceso de secuenciación

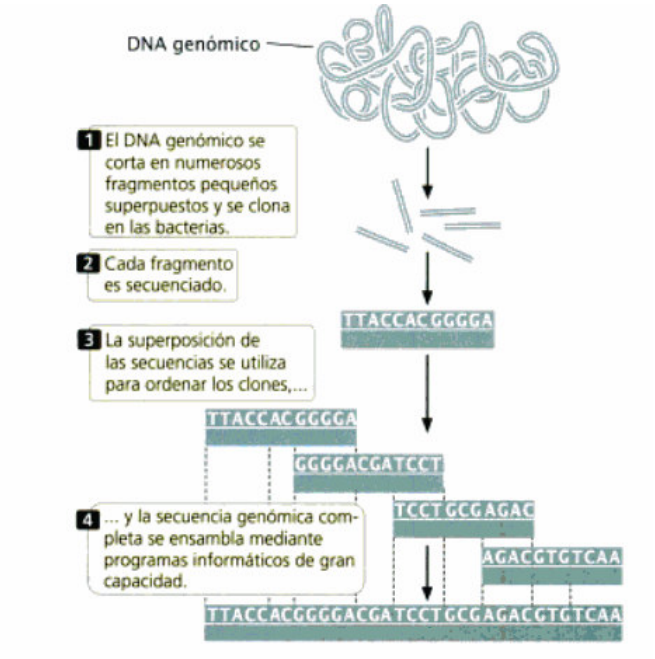


Figura 2.3: La secuenciación del genoma completo por fragmentos escogidos al azar (shotgun) utiliza frecuencias superpuestas para alinear los fragmentos. Se puede encontrar en [9].

de los haplotipos suele requerir la corrección de los datos iniciales. Así, el problema de haplotipaje de un individuo puede definirse de manera informal de la siguiente forma:

> *Dado material genético inconsistente procedente de la secuenciación del cromosoma de un individuo, encuentra y corrige los errores en los datos, dando lugar a un par de haplotipos consistente.*

Existen muchas versiones distintas de este problema, dependiendo del tipo de errores que se pretendan corregir. Entre los problemas de optimización de haplotipaje de individuos más destacados se incluyen el problema de eliminación del mínimo número de fragmentos y el problema de eliminación del mínimo número de SNP.

2.2.2. Haplotipaje de Poblaciones

Como hemos visto anteriormente, hoy en día existen procesos asequibles que permiten determinar información más genérica que los haplotipos a partir del ADN de los cromosomas. Esta se conoce como *genotipo*, y contiene información sobre los dos alelos de cada SNP, pero no permite conocer con qué alelos ha contribuido cada progenitor.

El problema de haplotipaje de poblaciones más genérico se puede enunciar como: *Dado un conjunto* $\mathfrak{G}$ *de genotipos, encuentra un conjunto* $\mathfrak{H}$ *de haplotipos que den lugar al conjunto* $\mathfrak{G}$.

Es fácil observar que, para un problema enunciado de esta forma, hay muchas soluciones posibles: por ejemplo, asumiendo que un individuo solo es heterocigótico en dos

SNPs, siendo los alelos {C,P} en el SNP 1 y {A,G} en el SNP 2, tenemos dos soluciones alternativas:

1. Un progenitor ha contribuido con los alelos T y A, y el otro progenitor con los alelos C y G.

2. Un progenitor ha contribuido con los alelos T y G, y el otro progenitor con los alelos C y A.

Por tanto, es claro que para k SNPs heterocigóticos, hay 2^k formas posibles de asociar los alelos a los padres. Tener tantas posibilidades convierte en difícil al problema de encontrar los haplotipos correctos. Cabe señalar que en este tipo de problemas, nos es indiferente a qué progenitor asociemos cada haplotipo, ya que no diferenciamos entre haplotipos paternos o maternos.

¿Cuál podemos considerar, entonces, un "buen" haplotipaje de poblaciones? Se han propuesto diversos criterios, todos ellos basados en motivaciones biológicas. Uno de los ampliamente aceptados es el Criterio de la Parsimonia, cuya idea subyacente es que "ante muchas posibles explicaciones de un fenómeno observado, aquella que resulta más simple suele ser la más probable". Aplicado en este contexto, una "buena" solución sería aquella que minimizara el número de haplotipos distintos que se obtienen. Este problema se conoce como el *Problema de Haplotipaje de la Parsimonia Pura* (HPP), y tiene como objetivo, dada una población de genotipos, minimizar el número de haplotipos que dan lugar a dicha población.

Una motivación biológica para utilizar este criterio es que el número de haplotipos distintos observados en la naturaleza es ampliamente inferior al número de haplotipos posibles. Además, como descendemos de un reducido número de ancestros, sus haplotipos han de ser, en teoría, los mismos que poseemos hoy en día (excluyendo los procesos de recombinación genética y las mutaciones). Este problema se estudia en profundidad en el capítulo siguiente.

Capítulo 3

El Problema de Haplotipaje de poblaciones de la Parsimonia Pura

Una vez realizada la contextualización del problema y dada la motivación biológica, pasaremos a estudiar el problema de Haplotipaje de la Parsimonia Pura en mayor profundidad. Para ello, haremos un recorrido a través de los distintos modelos de Optimización Lineal Entera que han sido propuestos, tanto de tamaño exponencial como polinómico. La primera formulación lineal entera para abordar el problema HPP fue propuesta por Gusfield en 2003, da lugar al Modelo TIP y se puede encontrar en [2]. Veremos, posteriormente, cómo puede reducirse el tamaño del Modelo TIP, dando lugar al Modelo RTIP, que permite resolver instancias más grandes del problema HPP. La segunda de las formulaciones estudiadas es de 2009, y se basa en una condición de cobertura de conjuntos, dando lugar al Modelo SC. Al igual que el Modelo TIP, el Modelo SC tiene tamaño exponencial, y fue propuesto por Lancia y Serafini en [5]. Por último, veremos una formulación propuesta en 2010 por Catanzaro, Godi y Labbé en [1] que tiene tamaño polinómico y está basada en el concepto de representante de clases. Primero introduciremos un Modelo Básico, y posteriormente veremos cómo reducir su tamaño fijando variables y utilizando distintas propiedades, lo que dará lugar al Modelo Reducido. A continuación, daremos algunas variaciones y mejoras del Modelo Reducido que no se han propuesto en [1] y que surgen a raíz de su análisis y posterior implementación en el programa de optimización Xpress. Tras esto, se estudian algunas desigualdades válidas propuestas en [1] para reforzar el Modelo Reducido, y se incluye un apartado con un estudio computacional comparativo de los modelos Básico y Reducido, así como del Reducido incluyendo algunos conjuntos de desigualdades válidas propuestas.

A continuación, damos la notación que utilizaremos para la descripción de los modelos y algunas definiciones que emplearemos a lo largo del capítulo. Estos contenidos se pueden encontrar en [4] y [5].

3.1. Notación y deficiones

Dado un SNP, el alelo más frecuente dentro de la población se cifra con el valor 0, y el menos frecuente, con el valor 1. Los haplotipos se representan, por tanto, como vectores

```
Individuo 1, paterno:   cagtacgtAcga ... tgatttatGatc ... ggatCtg ... catcatgatggtGagcta
Individuo 1, materno:   cagtacgtTcga ... tgatttatGatc ... ggatGtg ... catcatgatggtCagcta

Individuo 2, paterno:   cagtacgtAcga ... tgatttatCatc ... ggatCtg ... catcatgatggtCagcta
Individuo 2, materno:   cagtacgtAcga ... tgatttatGatc ... ggatCtg ... catcatgatggtCagcta

Individuo 3, paterno:   cagtacgtTcga ... tgatttatGatc ... ggatCtg ... catcatgatggtGagcta
Individuo 3, materno:   cagtacgtAcga ... tgatttatGatc ... ggatCtg ... catcatgatggtCagcta
```

Figura 3.1: Fragmento de un cromosoma en 3 individuos. Observamos que hay 4 SNP.

binarios de longitud n igual al número de SNPs.

En la Figura 3.1, fijamos en el primer SNP A como 0 y T como 1, en los otros tres SNP C como 0 y G como 1. Entonces, para el Individuo 1, el haplotipo paterno es el vector (0101) y el materno es el (1110).

Como para cada SNP tenemos tres posibilidades en un individuo (homocigótico de tipo 0 o 1, o heterocigótico), un genotipo se puede cifrar como un vector que toma valores en $\sum = \{0, 1, 2\}$, donde las entradas iguales a 0 (o 1) denotan posiciones homocigóticas de tipo 0 (o 1), mientras que las entradas iguales a 2 representan lugares heterocigóticos:

$$g_{kp} := \begin{cases} 0 & \text{si} \quad h_{ip} = h_{jp} = 0, \\ 1 & \text{si} \quad h_{ip} = h_{jp} = 1, \\ 2 & \text{si} \quad h_{ip} \neq h_{jp}. \end{cases}$$

Así, dado el par de haplotipos h_i, h_j, se define el operador suma $\oplus$ de h_i, h_j como el genotipo g_k cuya p-ésima entrada g_{kp} vale h_{ip} si $h_{ip} = h_{jp}$, y 2 si $h_{ip} \neq h_{jp}$. En el ejemplo de la Figura 3.1, dado por el haplotipo paterno (0100) y el materno (0101), el genotipo resultante sería el vector (0102).

Decimos que h_i, h_j *resuelven* un genotipo g_k si $g_k = h_i \oplus h_j$, y en tal caso decimos que h_i, h_j son *compatibles* con g_k. De forma similar, diremos que dos genotipos g_{k_1}, g_{k_2} son *compatibles* si existe un haplotipo h_k compatible con ambos (i.e., si $g_{k_{1p}} = g_{k_{2p}} \forall p \in \mathscr{P}$: $g_{k_{1p}} \neq 2 \neq g_{k_{2p}}$), y son *incompatibles* en caso contrario.

Llamaremos *posición ambigua* de g_k a aquella que contiene un 2, y $A(g_k)$ al conjunto de posiciones ambiguas de un genotipo g_k.

Haciendo uso de esta notación, los datos de entrada de un problema HPP consisten en un conjunto G de m genotipos, $g_1, ..., g_m$, correspondientes a m individuos de una población. Los datos de salida son un conjunto H de (como máximo $2m$) haplotipos y, para cada genotipo $g_k \in G$, un par de haplotipos $\{h_i, h_j\}$ que lo resuelven (es decir, tales que $g_k = h_i \oplus h_j$). A la hora de analizar los modelos de este capítulo, supondremos, sin pérdida de generalidad, que todos los genotipos de G son distintos y que en todos ellos hay al menos una posición ambigua.

Problema de Haplotipaje de la Parsimonia Pura (HPP) *Dado un conjunto de genotipos* $\mathfrak{G}$ *, encuentra un conjunto* $\mathfrak{H}$ *de haplotipos de cardinal mínimo que explique* $\mathfrak{G}$*.*

3.2. Modelos de Optimización Lineal Entera

En esta sección vamos a hacer un recorrido cronológico a través de diversas formulaciones de modelos de Optimización Lineal Entera que se han propuesto en las dos últimas décadas para abordar el problema de Haplotipaje de la Parsimonia Pura.

3.2.1. El Modelo TIP

Esta formulación fue propuesta por Gusfield en 2003 en [2], y es el primer modelo de Optimización Lineal Entera propuesto para abordar el problema de Haplotipaje de Poblaciones mediante el criterio de la Parsimonia Pura. Concretamente, en [2] se detalla un modelo de Optimización Lineal Entera que resuelve el problema HPP, aunque como el tiempo necesario se incrementa de manera exponencial con respecto al tamaño del problema, la formulación es válida para conjuntos de hasta 30 SNP y 50 genotipos, con niveles de heterocigosidad relativamente pequeños. Primero describimos un modelo conceptual, y posteriormente, una mejora que reduce considerablemente el número de variables de la formulación.

La Formulación TIP Conceptual

Comenzaremos describiendo una solución *conceptual* al problema HPP mediante una formulación de optimización lineal entera. Esta solución se dice conceptual porque, en general, será imposible de llevar a la práctica computacionalmente, pero con algunas modificaciones posteriores se podrá aplicar para resolver instancias de interés biológico.

Primero, vamos a dar las variables de decisión que emplearemos en la formulación del problema. Dado un genotipo g_i, supongamos que tiene Ai posiciones ambiguas. Entonces, existen 2^{Ai-1} parejas de haplotipos que podrían haber generado dicho genotipo. Llamaremos y_{ij} a las variables binarias que representan cada una de las 2^{Ai-1} parejas, donde $y_{ij} = 1$ si esa pareja se utiliza en la solución del problema, y 0 en caso contrario, numerando también los haplotipos que intervienen en dichas parejas. Para cada haplotipo que forme parte de una pareja y que no haya aparecido anteriormente, crearemos una variable x_i binaria. Así, solo se creará una variable x_i para cada haplotipo, independientemente del número de veces que forme parte de una pareja y_{ij}. x_i será igual a 1 cuando el haplotipo que representa se use en la solución para explicar alguno de los genotipos dados.

Pasamos ahora a dar las restricciones que componen la formulación.

De entre cada conjunto de parejas y_{ij} de haplotipos que explican un genotipo g_i, debemos seleccionar exactamente una de ellas en una solución factible del problema, luego la primera restricción que obtendremos será:

$$\sum_{j \in \{1,\dots,2^{Ai-1}\}} y_{ij} = 1, \qquad \forall i : g_i \in \mathfrak{G}, \tag{3.1}$$

Ahora supongamos que x_k, x_l son los haplotipos que componen una pareja y_{ij}. Entonces, claramente, si hemos seleccionado dicha pareja en nuestra solución (es decir,

$$
\begin{aligned}
\text{mín} \quad & \sum_{x \in \mathbf{X}} x \\
s.a \quad & \sum_{j \in \{1,\ldots,2^{Ai}-1\}} y_{ij} = 1, \qquad \forall i : g_i \in \mathfrak{G} \\
& y_{ij} \leq x_k \\
& y_{ij} \leq x_j \\
& y_{ij}, x_k \in \{0,1\}.
\end{aligned}
$$

Figura 3.2: Modelo TIP.

$y_{ij} = 1$), necesitamos que se incluyan los haplotipos en el conjunto de soluciones, es decir, $x_i = x_j = 1$. Así, para cada $y_{i,j}$ tendremos dos desigualdades:

$$y_{ij} \leq x_k \tag{3.2}$$
$$y_{ij} \leq x_j. \tag{3.3}$$

Por último, hemos de añadir las restricciones necesarias para que las variables y, x sean todas binarias:

$$y_{ij}, x_k \in \{0,1\}. \tag{3.4}$$

Si denotamos por $\mathbf{X}$ al conjunto de variables x generadas, hemos visto que existe una variable x por cada haplotipo distinto, y como queremos seleccionar el menor número posible, es claro que la función objetivo será:

$$\text{mín} \sum_{x \in \mathbf{X}} x \tag{3.5}$$

El conjunto de variables x que valen 1 en una solución óptima del problema constituyen una solución del problema HPP. La formulación completa del Modelo TIP se puede observar en la Figura 3.2.

El modelo RTIP

El número de desigualdades y variables requeridas en la formulación del modelo TIP hace que sea imposible emplearlo en la resolución de muchos problemas, por lo que se incluye a continuación una idea adicional que lo hace práctico para muchos más casos.

La idea es la siguiente: supongamos que existe una pareja de haplotipos h_i, h_j que explican un genotipo, y por tanto existe la variable y_{ij}, y que ninguno de los dos haplotipos puede explicar a algún otro genotipo del problema. Entonces no es necesario incluir en el problema las variables x_i, x_j e y_{ij}: si se eliminan todas las variables y asociadas a un genotipo, existirá una solución óptima del problema en la que la pareja de haplotipos de dicho genotipo se puede escoger arbitrariamente (de entre todas las que lo expliquen); y en otro caso, existirá una solución óptima del problema TIP original en la que ninguna

de las variables x o y eliminadas vale 1. Este modelo reducido, al que llamaremos RTIP, encontrará, por lo tanto, la misma solución óptima que el modelo TIP anterior.

Nos encontramos ahora una cuestión a tener en cuenta, y es que si primero enumeramos todos los haplotipos de la formulación, y luego realizamos el proceso anterior para eliminar variables, el trabajo que ello conlleva podría hacer que el modelo RTIP fuera también impracticable, ya que la enumeración de tal cantidad de variables consumiría gran cantidad de tiempo. Planteamos entonces una forma de incluir en la formulación los haplotipos que sabemos que son compatibles con al menos dos genotipos. Dados $g_1, g_2 \in \mathfrak{G}$, sean H_1, H_2 los conjuntos de haplotipos compatibles con cada genotipo, y $H_1 \cap H_2$ el conjunto de haplotipos compatibles con ambos. Si para un SNP p, $g_{1p} = 0$ y $g_{2p} = 1$ (o viceversa), entonces los genotipos g_1, g_2 son incompatibles y $|H_1 \cap H_2| = 0$; si $g_{1p} = 2$ y $g_{2p} \in \{0, 1\}$ (o viceversa), entonces cualquier haplotipo $h \in H_1 \cap H_2$ será tal que $h_p = g_{2p}$; y por último, si $g_{1p} = g_{2p} = 2$, entonces en la posición p un haplotipo compatible con g_1, g_2 podrá tener un 0 o un 1. Por tanto, es claro que el cardinal de $H_1 \cap H_2$ vendrá dado por las posiciones ambiguas de ambos genotipos: si existen k posiciones ambiguas en los genotipos g_1, g_2, entonces $|H_1 \cap H_2| = 2^k$, y los generaremos fijando estas k posiciones a 0 o 1 de todas las formas posibles. Repitiendo este proceso con todas las parejas de genotipos de $\mathfrak{G}$, habremos incluido en la formulación los haplotipos del Modelo RTIP.

3.2.2. El Modelo SC

Otra de las formulaciones de Optimización Entera propuestas para abordar el problema HPP es el un problema de cubrimiento de conjuntos (en inglés *Set Covering*, abreviado SC). Este modelo fue propuesto por Lancia y Serafini en 2009, y se puede encontrar en [5]. El Modelo SC también cuenta con un número de variables y restricciones que crece de manera exponencial con respecto al tamaño del problema, pero en el que las variables y restricciones se van añadiendo dinámicamente a medida que son necesarias, permitiendo de esta forma abordar instancias que no era posible resolver con los modelos anteriores a este.

La idea principal del Modelo SC se basa en una condición que tiene que cumplir cualquier conjunto $\mathfrak{H}$ de haplotipos que resuelva a $\mathfrak{G}$, que es la siguiente:

> **Condición de cubrimiento:** Para cada genotipo $g \in \mathfrak{G}$, $p \in A(g)$ y $a \in \{0, 1\}$, existe un haplotipo $h \in \mathfrak{H}$ compatible con g tal que $h_p = a$.

Esta es una condición necesaria, pero no suficiente, para que $\mathfrak{H}$ sea solución factible del problema HPP. Para comprobar que no es suficiente, observemos el siguiente conjunto de genotipos:

$$\mathfrak{G}' = \begin{cases} 0\ 2\ 2\ 2 \\ 2\ 0\ 2\ 2 \\ 2\ 2\ 0\ 2 \\ 2\ 2\ 2\ 0 \end{cases}$$

Para $\mathfrak{G}'$, el conjunto $\{h_1 = (1000), h_2 = (0100), h_3 = (0010), h_4 = (0001)\}$ satisface la condición de cubrimiento, pero no es un conjunto factible para el problema HPP. En efecto,

vemos que para el genotipo (0222) y la posición ambigua dada por el SNP 2, h_2 cumple $a = 1$ y es compatible con (0222), mientras que h_3 cumple $a = 0$ y también es compatible con dicho genotipo, y el resto de posiciones ambiguas están cubiertas por la condición de cubrimiento siguiendo un razonamiento análogo. No obstante, no encontramos en $h_i, i \in \{1, ... ,4\}$ dos genotipos tales que $h_i \oplus h_j = (0222)$.

Esta condición se llama de cubrimiento porque, definiendo el conjunto $C = \{(g, p, a) \mid g \in \mathfrak{G}, p \in A(g), a \in \{0, 1\}\}$, podemos identificar un haplotipo h con un subconjunto del anterior $C_h = \{(g, p, a) \in C \mid h \in \mathfrak{H}, h \text{ es compatible con } g, \text{ y } h_p = a\}$. Así, la condición de cubrimiento implica que H cubra el conjunto C.

En este contexto, tiene sentido considerar el problema de minimizar el número de haplotipos necesarios para satisfacer la condición de cubrimiento. De hecho, en base a lo anterior, este problema es una relajación del problema HPP, pero podemos encontrar una solución óptima de HPP añadiendo algunas restricciones adicionales que veremos posteriormente.

Introducimos ahora las variables que utilizaremos en la formulación de un modelo que minimiza el número de haplotipos necesarios para cumplir la condición de cubrimiento en un conjunto $\mathfrak{G}$ de genotipos. Sea g un genotipo, y $p \in A(g)$ una posición ambigua de g. Para $a \in \{0, 1\}$, definimos el conjunto $H_p^a(g)$ como el conjunto de haplotipos compatibles con g que tienen el valor a en el SNP número p. Es claro que los conjuntos H_p^0, H_p^1 forman una partición en el conjunto de haplotipos compatibles con g, y que si $A(g) = k$, entonces $|H_p^0| = |H_p^1| = 2^{k-1}$. Para cada haplotipo h compatible con un $g \in \mathfrak{G}$, introducimos la variable binaria x_h, que valdrá 1 si h pertenece al conjunto solución, y 0 en caso contrario. Como cada x hace referencia a un haplotipo, podremos referirnos en ocasiones a la variable x_i como el haplotipo i.

Vistas las variables binarias que componen el modelo, damos a continuación sus restricciones. Para empezar, imponemos que todas las variables sean binarias con el conjunto de restricciones

$$x_h \in \{0, 1\} \ \forall h \in H_{\mathfrak{G}},$$

donde $H_{\mathfrak{G}}$ es el conjunto de haplotipos compatibles con algún genotipo de $\mathfrak{G}$.

Además, incluimos un conjunto de restricciones para garantizar que se cumple la condición de cubrimiento:

$$\sum_{h \in H_p^a(g)} x_h \geq 1 \qquad \forall g \in \mathfrak{G}, p \in A(g), a \in \{0, 1\}.$$

Por último, damos el objetivo del problema, que será minimizar el número de haplotipos que cumplan las restricciones:

$$\text{mín} \sum_{h \in H_{\mathfrak{G}}} x_h.$$

Aunque, según [5], la cota inferior que proporciona el problema es fuerte, es decir, no está muy alejada del óptimo que proporciona el HPP, vamos a ver cómo introducir restricciones que violan soluciones enteras del problema de cubrimiento que son infactibles para el HPP.

$$\text{mín} \quad \sum_{h \in H_{\mathfrak{G}}} x_h \tag{3.6}$$

$$s.a \quad \sum_{h \in H_p^0(g)} x_h \geq 1 \qquad \forall g \in \mathfrak{G}, p \in A(g) \tag{3.7}$$

$$\sum_{h \in H_p^1(g)} x_h \geq 1 \qquad \forall g \in \mathfrak{G}, p \in A(g) \tag{3.8}$$

$$x(C(g, H^{'})) \geq 1 \qquad \forall (g, H^{'}) \in N^{'}. \tag{3.9}$$

$$x_h \in \{0, 1\} \qquad \forall h \in H(g). \tag{3.10}$$

Figura 3.3: Modelo SC.

Llamaremos a un conjunto $H^{'}$ de haplotipos *insuficiente* si no resuelve a $\mathfrak{G}$. Dado $H^{'}$ conjunto insuficiente, sea $U(H^{'})$ el conjunto de genotipos de $\mathfrak{G}$ no resueltos por $H^{'}$. Para $g \in U(H^{'})$, llamamos $C(g, H^{'}) = H(g) - H^{'}$. Un genotipo $g \in U(H^{'})$ da lugar a la siguiente restricción:

$$x(C(g, H^{'})) \geq 1.$$

Llamando $N^{'}$ al conjunto de todos los pares $(g, H^{'})$ con $H^{'}$ un conjunto insuficiente y $g \in U(H^{'})$, obtenemos la formulación completa del Modelo SC, que se puede observar en la Figura 3.3.

Esta formulación tiene un número exponencial de variables y restricciones, por lo que, para poder hacer uso de ella, se mantiene el modelo con las restricciones (3.7) y (3.8), y el conjunto de restricciones (3.9) se actualiza durante el proceso de resolución. Este proceso, que se realiza por el algoritmo de ramificación y acotación, incluye la resolución de la relajación lineal del problema y su implementación, y no se detalla aquí. El lector interesado puede encontrarlo en [5].

3.2.3. El Modelo Básico

Tanto este modelo como el Modelo Reducido que se introduce en el apartado siguiente fueron propuestos por Catanzaro, Godi y Labbé en 2010, y se pueden encontrar en [1]. Estos modelos proporcionan mejores resultados que los modelos RTIP y SC vistos anteriormente, ya que el número de variables y restricciones que intervienen crece de forma polinómica con respecto al tamaño del problema a abordar. Como veremos posteriormente, el Modelo Reducido, que parte del Modelo Básico pero explota las características del problema para reducir su tamaño, puede resolver instancias de 100 SNP y 100 genotipos, es decir, de tamaño mucho mayor que los modelos propuestos con anterioridad.

Antes de dar la formulación del modelo, vamos a estudiar algunas propiedades que comparten las soluciones factibles del problema HPP y que nos servirán para explicar las variables de decisión que introduciremos después.

Genotipos	SNP			
Genotipo 1	2	2	2	2
Genotipo 2	1	2	0	2
Genotipo 3	0	2	1	2

Tabla 3.1: Ejemplo de un problema HPP.

Haplotipos	SNP			
Hapl. 1	1	0	0	0
Hapl. 2	0	0	1	1
Hapl. 3	1	0	0	1
Hapl. 4	0	1	1	0

Resolución de los genotipos				
Gen. 1	=	Hapl. 1	$\oplus$	Hapl. 2
Gen. 2	=	Hapl. 1	$\oplus$	Hapl. 3
Gen. 3	=	Hapl. 2	$\oplus$	Hapl. 4

Tabla 3.2: Solución óptima 1.

Haplotipos	SNP			
Hapl. 1	1	0	0	0
Hapl. 2	0	0	1	1
Hapl. 3	1	0	0	1
Hapl. 4	0	1	1	0

Resolución de los genotipos				
Gen. 1	=	Hapl. 3	$\oplus$	Hapl. 4
Gen. 2	=	Hapl. 1	$\oplus$	Hapl. 3
Gen. 3	=	Hapl. 2	$\oplus$	Hapl. 4

Tabla 3.3: Solución óptima 2.

Dada una solución factible de un problema HPP, esta se puede representar mediante un grafo bipartito en el que los vértices son los genotipos dados por el problema y los haplotipos de la solución factible, y cada vértice g_k dado por un genotipo es adyacente a los dos haplotipos h_i, h_j que lo resuelven (y, por tanto, tiene grado dos). En la Tabla 3.1 tenemos un ejemplo de un problema sencillo para el que tenemos dos soluciones óptimas alternativas, que podemos encontrar en las Tablas 3.2 y 3.3. En la Figura 3.4 se muestran sus correspondientes grafos bipartitos construidos de la forma descrita anteriormente. En ella, podemos observar que los vértices correspondientes a genotipos tienen grado dos, ya que son adyacentes a los vértices de los haplotipos que los explican.

En esta representación mediante un grafo bipartito de una solución, cada haplotipo es adyacente a un conjunto de genotipos (que serán aquellos explicados por dicho haplotipo). La familia de subconjuntos de genotipos inducidos por cada haplotipo satisface tres propiedades:

1. Claramente, cada subconjunto de genotipos está explicado por el haplotipo que induce dicho subconjunto.

2. Como un genotipo se explica por dos haplotipos, cada genotipo pertenece a dos de estos subconjuntos.

3. Cada par de subconjuntos tienen en común, como máximo, a un único genotipo. Esto se deduce del hecho de que si dos genotipos pertenecen a los dos mismos subconjuntos, entonces están explicados por los dos haplotipos que inducen dichos subconjuntos, luego son iguales.

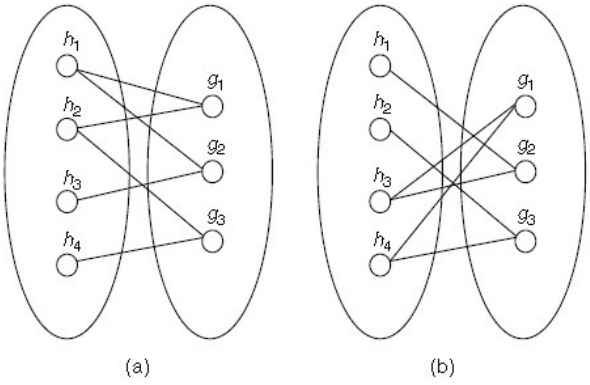


Figura 3.4: Grafos bipartitos formados a partir de las Tablas 3.2 y 3.3 de soluciones. Se puede encontrar en [1].

En el ejemplo de la Tabla 3.1, la solución 1 de la Tabla 3.2 induce el grafo bipartito de la Figura 3.4(a). Los subconjuntos inducidos por los cuatro haplotipos de la solución son: $S_1 = \{(2222), (1202)\}$, inducido por el haplotipo $h_1 = (1100)$; $S_2 = \{(2222), (0212)\}$, inducido por $h_2 = (0011)$; $S_3 = \{(1202)\}$, inducido por $h_3 = (1001)$; y $S_4 = \{(0212)\}$, inducido por $h_4 = (0110)$.

Utilizando las tres propiedades vistas anteriormente, asociamos a cada subconjunto S de genotipos adyacentes a un haplotipo en el grafo bipartito correspondiente un índice dado por el menor índice de los genotipos del conjunto. Específicamente, si el menor de los genotipos es g_i, entonces el subconjunto será denotado por S_i. Como cada genotipo pertenece a dos de estos subconjuntos, existe la posibilidad de que un genotipo g_j sea el menor en dos subconjuntos distintos. En este caso, un subconjunto se denotará por S_j y el otro por $S_{j'}$. Es importante señalar que $S_{j'}$ solo podrá ser creado si S_j existe previamente. A partir de ahora, denotaremos por *el genotipo k* al genotipo g_k.

Pasamos ahora a introducir las variables que utilizaremos para el Modelo Básico. Dados m genotipos, sabemos que el número máximo de haplotipos que podremos necesitar para resolverlos es $2m$, por lo que los índices i en los que se mueven los subconjuntos S_i variarán en el conjunto $K \cup K^{'}$, donde $K = \{1, ..., m\}$ y $K^{'} = \{1^{'}, ..., m'\}$, con el orden $1 < 1^{'} < 2 < 2^{'} < ... < m < m^{'}$. Por último, denotamos por $\mathscr{P}$ al conjunto de p SNP que caracterizan cada genotipo g_k de $\mathfrak{G}$, y g_{kp} denotará la posición del SNP p en el genotipo k. Las variables de decisión que vamos a utilizar son:

- Llamamos x_i, $\forall i \in K \cup K^{'}$, a una variable binaria que será igual a 1 si en la solución existe un haplotipo que induce el subconjunto S_i de genotipos, y 0 en otro caso.
- Denotamos por y_{ij}^k, $\forall k \in K, \forall i, j \in K \cup K^{'}, i < j$, a una variable binaria que valdrá 1 en la solución si el genotipo k pertenece a los conjuntos S_i, S_j, y 0 en otro caso.
- Por último, llamamos z_{ip}, $\forall i \in K \cup K^{'}, \forall p \in \mathscr{P}$, a una variable binaria que será igual a 1 si el haplotipo que induce el subconjunto S_i de genotipos tiene un 1 en la posición p, y valdrá 0 si tiene un 0 en dicha posición. Este conjunto de variables describe explícitamente el conjunto de haplotipos de la solución.

Utilizando esta notación, damos a continuación la formulación de un modelo de op-

timización Lineal Entera para el problema HPP, que llamaremos Modelo Básico. Como $x_i = 1$ implica que existe un haplotipo que induce S_i, es claro que la suma de los x_i nos da el número de haplotipos que se usan para explicar algún genotipo, es decir, el conjunto $\mathfrak{H}$. Entonces podemos expresar la función objetivo como

$$\text{mín} \sum_{i \in K \cup K'} x_i. \tag{3.11}$$

A continuación describimos las restricciones del modelo. Para considerar $S_{i'}$ solo si existe previamente S_i, imponemos:

$$x_{i'} \leq x_i \qquad \forall i \in K. \tag{3.12}$$

Además, cada genotipo k debe pertenecer a un par de subconjuntos S_i, S_j, y si k es resuelto por h_i, entonces debe existir el subconjunto inducido por dicho haplotipo, luego tenemos las restricciones:

$$\sum_{i,j \in K \cup K'} y_{ij}^k \geq 1 \qquad \forall k \in K, \tag{3.13}$$

$$\sum_{j \in K \cup K' : j \geq i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \leq x_i \qquad \forall k \in K, \forall i \in K \cup K^{'}. \tag{3.14}$$

Un caso particular del conjunto de restricciones (3.14) se tiene cuando uno de los haplotipos que resuelve a un genotipo es $k^{'}$. En este caso, sabemos que existe el conjunto $S_{k'}$, luego también existe S_k, y es claro que g_k pertenece a ambos subconjuntos, luego $y_{kk'}^k = 1$, y por tanto podemos forzar a la variable $x_{k'}$ a ser 1 mediante la restricción:

$$y_{kk'}^k \leq x_{k'} \qquad \forall k \in K. \tag{3.15}$$

Como las variables $x_{kp}, z_{k'p}, \forall p \in \mathscr{P}$ explican al genotipo g_k, tenemos que imponer tres restricciones necesarias para que $z_k \oplus z_{k'} = g_k$:

$$z_{kp} = z_{k'p} = 0 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 0, \tag{3.16}$$

$$z_{kp} = z_{k'p} = 1 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 1, \tag{3.17}$$

$$z_{kp} + z_{k'p} = 1 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 2. \tag{3.18}$$

Las siguientes cuatro restricciones relacionan las variables y_{ij}^k y z_{ip}, ya que imponen que para cada genotipo g_k perteneciente a S_i, h_i explique a g_k. En concreto, si existe un genotipo g_k en S_i tal que $g_{kp} = 0$ (o 1) para algún $p \in \mathscr{P}$, entonces $z_{kp} = 0$ (o 1), luego:

$$z_{ip} \leq 1 - \sum_{j \in K \cup K' : j \geq i} y_{ij}^k - \sum_{j \in K \cup K' : j < i} y_{ji}^k$$
$$\forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 0, \forall i \in K \cup K^{'}, i \neq k, k^{'}, \tag{3.19}$$

$$z_{ip} \geq \sum_{j \in K \cup K' : j \geq i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k$$
$$\forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 1, \forall i \in K \cup K^{'}, i \neq k, k^{''}, \tag{3.20}$$

$$\text{mín} \quad \sum_{i \in K \cup K'} x_i \tag{3.11}$$

$$s.a \quad x_{i'} \leq x_i \qquad \forall i \in K \tag{3.12}$$

$$\sum_{i,j \in K \cup K'} y_{ij}^k \geq 1 \qquad \forall k \in K \tag{3.13}$$

$$\sum_{j \in K \cup K' : j \geq i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \leq x_i \qquad \forall k \in K, \forall i \in K \cup K^{'} \tag{3.14}$$

$$y_{kk'}^k \leq x_{k'} \qquad \forall k \in K \tag{3.15}$$

$$z_{kp} = z_{k'p} = 0 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 0 \tag{3.16}$$

$$z_{kp} = z_{k'p} = 1 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 1 \tag{3.17}$$

$$z_{kp} + z_{k'p} = 1 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 2 \tag{3.18}$$

$$z_{ip} \leq 1 - \sum_{j \in K \cup K' : j \geq i} y_{ij}^k - \sum_{j \in K \cup K' : j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 0, \forall i \in K \cup K^{'}, i \neq k, k^{'} \tag{3.19}$$

$$z_{ip} \geq \sum_{j \in K \cup K' : j \geq i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 1, \forall i \in K \cup K^{'}, i \neq k, k^{''} \tag{3.20}$$

$$z_{ip} + z_{jp} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 2, \forall i, j \in K \cup K^{'} \tag{3.21}$$

$$z_{ip} + z_{jp} \leq 2 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 2, \forall i, j \in K \cup K^{'} \tag{3.22}$$

$$x_i, z_{ip}, y_{ij}^k \in \{0, 1\}. \tag{3.23}$$

Figura 3.5: Modelo Básico.

y del mismo modo, si $g_{kp} = 2$, entonces únicamente si $g_k \in S_i, S_j$, queremos $y_{ij}^k = 1 = z_{ip} + z_{jp}$, por lo que empleamos las restricciones:

$$z_{ip} + z_{jp} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 2, \forall i, j \in K \cup K^{'}, \tag{3.21}$$

$$z_{ip} + z_{jp} \leq 2 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K : g_{kp} = 2, \forall i, j \in K \cup K^{'}. \tag{3.22}$$

Por último, imponemos que las variables de decisión sean todas binarias:

$$x_i, z_{ip}, y_{ij}^k \in \{0, 1\}. \tag{3.23}$$

Podemos observar la formulación del Modelo Básico en la Figura 3.5.

Una vez hemos visto la formulación del Modelo Básico, veamos un ejemplo y su resolución. En la Figura 3.6 tenemos un conjunto $\mathfrak{G}^{'}$ de 5 genotipos y la solución que nos proporciona el Modelo Básico si implementamos este ejemplo en Xpress[1]. En él,

[1]El código del Modelo Básico utilizado se puede encontrar en el Anexo 1.

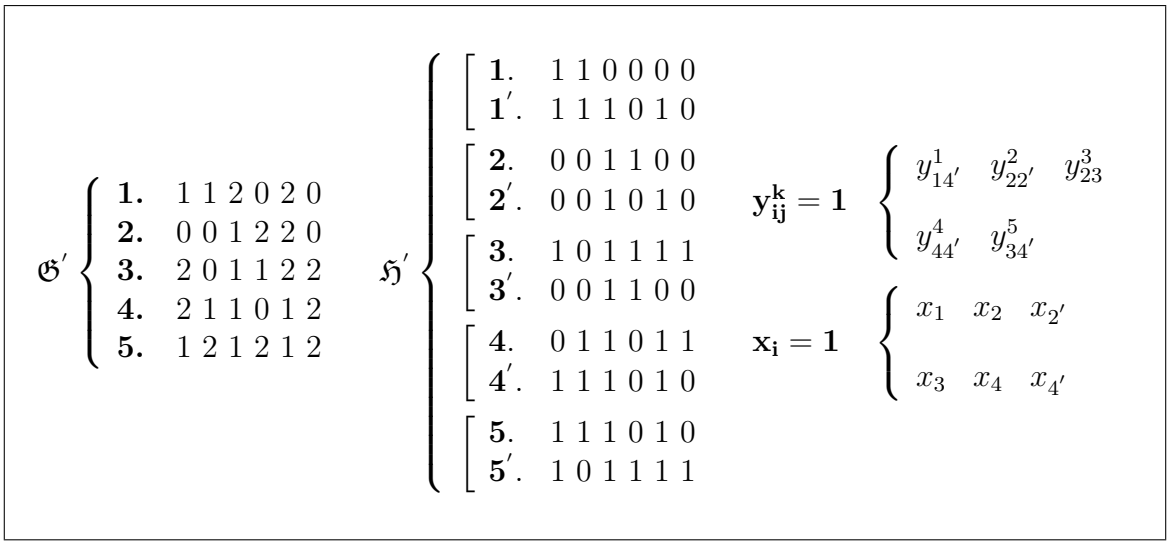


Figura 3.6: Ejemplo $\mathfrak{G}'$ y soluciones $\mathfrak{H}' = \{h_i\}$, $i \in K \cup K'$, $\mathbf{y_{ij}^k}$ y $\mathbf{x_i}$ dadas por el programa Xpress introduciendo el código del Modelo Básico.

podemos ver que se cumplen las restricciones (3.16)-(3.18) del Modelo Básico, ya que $h_i \oplus h_{i'} = g_i \; \forall i = \{1, ... ,5\}$. Por ejemplo, para el genotipo g_1, $h_{1p} \oplus h_{1'p} = g_{1p} \; \forall p \in \{1, ... ,6\}$. No obstante, según la solución del problema en base a los conjuntos y_{ij}^k, el haplotipo $h_{1'}$ no se emplea en la solución, sino que los dos haplotipos que resuelven a g_1 son $h_1, h_{4'}$ (porque $y_{14'}^1 = 1$). Si nos fijamos, comprobaremos que $h_{1'} = h_{4'}$, luego en efecto se cumple también $h_1 \oplus h_{4'} = g_1$. Esto es debido a la existencia de las restricciones (3.19)-(3.22), que fuerzan a que se cumpla $h_i \oplus h_j = g_k$ siempre que $y_{ij}^k = 1$. Del mismo modo, vemos que se cumple la restricción (3.13): cada genotipo está explicado por dos haplotipos, pues hay 5 conjuntos y_{ij}^k, con $k = \{1, ... ,5\}$, que valen 1 en la solución; y para cada haplotipo i (o j) tal que $y_{ij}^k = 1$, x_i (o x_j) vale 1, respetando las restricciones (3.14). Del mismo modo, vemos que si se emplea un haplotipo $x_{i'}$ en una solución, como en los casos $x_{2'}$ y $x_{4'}$, es porque primero se han utilizado los haplotipos x_2 y x_4, tal y como indica la restricción (3.12), y que en este caso, los conjuntos $y_{22'}^2$ y $y_{44'}^4$ valen 1, forzados por la restricción (3.15), ya que solo se podía emplear un conjunto $S_{i'}$ si previamente se había empleado S_i, es decir, cuando un genotipo era el menor de dos conjuntos.

Sin embargo, vemos que en esta solución del problema, aunque correcta, no se respetan todos los planteamientos iniciales previos: por ejemplo, el genotipo 1 está resuelto por los haplotipos 1 y $4'$, lo que significa que pertenece a los conjuntos S_1 y $S_{4'}$. Pero entonces el conjunto $S_{4'}$ contiene genotipos menores que $4'$, como el 1. Siguiendo este razonamiento, como el genotipo 1 siempre es el menor de dos conjuntos, los conjuntos S_1, $S_{1'}$, la variable $y_{11'}^1$ debería valer 1 en cualquier solución, y conjuntos del tipo $y_{14'}^k$, para $k = \{1, 2, 3\}$, siempre deberían valer 0 en una solución óptima. Este y otros razonamientos son los que motivan el siguiente apartado.

3.2.4. El Modelo Reducido

En este apartado, vamos a reducir el tamaño del Modelo Básico, dando lugar a una formulación mejorada. Para ello, vamos a centrarnos en eliminar algunas de las variables y_{ij}^k que, por la forma en que están definidos los subconjuntos S_i, sabemos que van a valer 0 en cualquier solución.

- Primero, recordemos que hemos eliminado de la formulación el conjunto de variables:

$$R_1 = \{y_{ij}^k : k \in K, i, j \in K \cup K^{'}, j \leq i\}. \tag{3.24}$$

- Como los subconjuntos S se nombran según el índice del menor de los genotipos que contienen, es evidente que un genotipo g_k nunca va a pertenecer a los conjuntos $S_i, S_j, i < j$ para $i > k$, y si $i \leq k < j$, y_{ij}^k puede ser 1 solo en el caso $k = i$, $k^{'} = j$. Por tanto, no necesitamos definir las variables de decisión del conjunto

$$R_2 = \{y_{ij}^k : k \in K, i, j \in K \cup K^{'}, i > k \ o \ j > k^{'} \ o \ i = k, j \neq k^{'}\}. \tag{3.25}$$

- Además, la variable $y_{ii'}^k$ solo puede valer 1 cuando $k = i$, y es claro que $y_{11'}^1 = 1$ siempre, luego no necesitamos definir las variables de:

$$R_{3_a} = \{y_{11'}^k : k \in K, k \neq 1\}, \tag{3.26}$$

$$R_{3_b} = \{y_{ii'}^k : k \in K, i \in K, 2 \leq i < k\}. \tag{3.27}$$

- Por último, una variable de la forma $y_{ik'}^k$ solo puede valer 1 si $i = k$, luego el siguiente conjunto es innecesario:

$$R_4 = \{y_{ik'}^k : k \in K, i \in K \cup K^{'}, i < k\}. \tag{3.28}$$

Podemos seguir ampliando el conjunto de variables innecesarias mediante las siguientes proposiciones:

Proposición 3.1. *El conjunto de variables*

$$R_5 = \{y_{ij}^k : i, j, k \in K, p \in \mathscr{P}, g_{kp} = 2, g_{ip} = g_{jp} \neq 2\} \tag{3.29}$$

es innecesario.

Demostración. Supongamos $g_{ip} = g_{jp} = 0$, $g_{kp} = 2$, g_k perteneciente a S_i, S_j (es decir, $y_{ij}^k = 1$). Entonces, como $g_k \in S_i$, se tiene que el haplotipo h_i explica a g_i y a g_k, luego h_i tiene un 0 en el SNP p. Del mismo modo, h_j tiene un 0 en la posición p. Pero entonces se viola la condición $h_i \oplus h_j = g_k$ en el SNP p, luego tenemos una contradicción. Por lo tanto, $y_{ij}^k = 0$. □

Proposición 3.2. *El conjunto de variables*

$$R_6 = \{y_{ij}^k : i, j, k \in K, p \in \mathscr{P}, g_{kp} + g_{ip} = 1 \ o \ g_{kp} + g_{jp} = 1\} \tag{3.30}$$

es innecesario.

Demostración. Supongamos, buscando una contradicción, que el conjunto R_6 no es redundante. Entonces tenemos que, para algún $p \in \mathscr{P}$ el genotipo k puede pertenecer a dos subconjuntos S_i, S_j. Sin pérdida de generalidad, supongamos $g_{kp} = 0$ y $g_{ip} = 1$. Como el haplotipo h_i explica al genotipo g_i, debe tener un 1 en el SNP de posición p, pero esto implica que h_i no puede explicar a g_k, ya que una condición necesaria para ello es que su SNP p sea 0. Por tanto, $y_{ij}^k = 0$. El resto de casos se razonan de forma análoga. □

Empleando lo anterior, vamos a dar la formulación de un Modelo Reducido. Denotamos por Υ al conjunto de variables y_{ij}^k, $R = \cup_q R_q$, con $q = \{1, 2, 3_a, 3_b, 4, 5, 6\}$, y $\hat{\Upsilon} = \Upsilon \backslash R$.

La función objetivo del Modelo Reducido es la misma que la del Modelo Básico:

$$\text{mín} \sum_{i \in K \cup K'} x_i. \tag{3.31}$$

La restricción necesaria para que solo se creen los subconjuntos $S_{i'}$ si existen previamente los S_i, tampoco varía respecto del modelo anterior:

$$x_{i'} \leq x_i \qquad \forall i \in K. \tag{3.32}$$

Por su parte, las restricciones (3.13) y (3.15) del Modelo Básico no varían en el nuevo modelo, salvo por que ahora los y_{ij}^k se mueven en $\hat{\Upsilon}$:

$$\sum_{i,j: y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k \geq 1 \qquad \forall k \in K, \tag{3.33}$$

$$\sum_{i,j: y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k + \sum_{i,j: y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \leq x_i \qquad \forall k \in K, \forall i : y_{ij}^k \in \hat{\Upsilon}, \tag{3.34}$$

$$y_{kk'}^k \leq x_{k'} \qquad \forall k \in K. \tag{3.35}$$

De entre los conjuntos de restricciones (3.16)-(3.18), los dos primeros no son necesarios en la nueva formulación, de modo que incluimos solo el último:

$$z_{kp} + z_{k'p} = 1 \qquad \forall k \in K, \forall p \in \mathscr{P} : g_{kp} = 2. \tag{3.36}$$

Pasamos ahora a explicar otros conjuntos de restricciones que sustituyen a las restricciones (3.19)-(3.22) del Modelo Básico. Recordemos que, si $g_k \in S_i, S_j$, entonces g_k está explicado por los haplotipos h_i, h_j que inducen dichos subconjuntos, y estos haplotipos vienen dados por las variables $z_{ip}, z_{jp}, \forall p \in \mathscr{P}$. Los siguientes conjuntos de restricciones se incluyen, por tanto, para forzar a que, si el genotipo g_k pertenece a los subconjuntos S_i, S_j en una solución factible, entonces los haplotipos h_i, h_j que inducen a los subconjuntos expliquen a g_k, es decir, que $z_{ip} \oplus z_{jp} = g_{kp}, \forall p \in \mathscr{P}$. A continuación vamos a explicar en qué situaciones son necesarios cada uno de estos conjuntos de restricciones:

- Supongamos que existe un genotipo g_k que pertenece a S_i, S_j, con $g_{kp} = 0$, $g_{ip} = 2$ para algún $p \in \mathscr{P}$. Por estar g_k en S_i, S_j, se tiene que cumplir $z_{ip} \oplus z_{jp} = g_{kp} = 0$,

y por tanto queremos $z_{ip} = 0$. De forma análoga, si suponemos $g_{kp} = 1$, $g_{ip} = 2$, buscamos $z_{ip} = 1$, lo que nos lleva a las restricciones:

$$z_{ip} \leq 1 - \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k - \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i : y_{ij}^k \in \hat{\Upsilon}, g_{kp} = 0, g_{ip} = 2, \tag{3.37}$$

$$z_{ip} \geq \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k + \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i : y_{ij}^k \in \hat{\Upsilon}, g_{kp} = 1, g_{ip} = 2. \tag{3.38}$$

- Supongamos ahora que existe un genotipo g_k perteneciente a S_i, S_j, con $g_{kp} = 2$, $g_{ip} = 2$, $g_{jp} = 0$ para un $p \in \mathscr{P}$. Como el haplotipo h_j explica a g_j (porque $g_j \in S_j$), $z_{jp} = 0$. Pero entonces para que se cumpla $z_{ip} \oplus z_{jp} = g_{kp}$, hemos de forzar a que $z_{ip} = 1$. Esto nos lleva a las restricciones:

$$z_{ip} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 0, g_{kp} = 2. \tag{3.39}$$

- De manera análoga, si existe un genotipo g_k perteneciente a S_i, S_j, con $g_{kp} = 2$, $g_{ip} = 0$, $g_{jp} = 2$ para un $p \in \mathscr{P}$, entonces tenemos que $z_{ip} = 0$ y debemos forzar que $z_{jp} = 1$, luego obtenemos:

$$z_{ip} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 0, g_{jp} = 2, g_{kp} = 2. \tag{3.40}$$

- Si suponemos que tenemos $g_k \in S_i, S_j$ con $g_{kp} = 2$, $g_{ip} = 2$, $g_{jp} = 1$ para un $p \in \mathscr{P}$, entonces $z_{jp} = 1$, y buscaremos una restricción que haga $z_{ip} = 0$, luego obtenemos:

$$z_{ip} \leq 1 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 1, g_{kp} = 2. \tag{3.41}$$

- Si suponemos, de forma análoga a la anterior, que tenemos $g_k \in S_i, S_j$ con $g_{kp} = 2$, $g_{ip} = 1$, $g_{jp} = 2$ para un $p \in \mathscr{P}$, tendremos que $z_{ip} = 1$ porque h_i explica a g_i, y buscaremos $z_{jp} = 0$, luego:

$$z_{ip} \leq 1 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 1, g_{jp} = 2, g_{kp} = 2. \tag{3.42}$$

- Por último, supongamos que tenemos $g_k \in S_i, S_j$ con $g_{kp} = 2$, $g_{ip} = 2$, $g_{jp} = 2$ para un $p \in \mathscr{P}$. En este caso, sabemos que h_i explica a g_i, g_k, mientras que h_j explica a g_j, g_k, y como $g_{kp} = 2$, tiene que cumplirse $z_{ip} \oplus z_{jp} = 2$ luego $z_{ip} \neq z_{jp}$, y por tanto su suma vale 1. En este caso, son necesarios los siguientes conjuntos de restricciones:

$$z_{ip} + z_{jp} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K,$$
$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2, \tag{3.43}$$

$$z_{ip} + z_{jp} \leq 2 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K, \\ \forall i,j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2. \tag{3.44}$$

Finalmente, completamos las restricciones exigiendo que las variables sean binarias:

$$x_i, z_{ip}, y_{ij}^k \in \{0,1\}. \tag{3.45}$$

Es fácil observar que la integridad de las variables y_{ij}^k garantiza la de las variables x_i en el óptimo, por lo que podemos relajar la condición de integridad de estas últimas.

3.2.5. Algunas mejoras del Modelo Reducido

A raíz del estudio en profundidad y la implementación en el programa de optimización Xpress de la formulación del Modelo Reducido dado en [1] y descrito en la sección anterior, nos han surgido algunas correcciones y mejoras, no propuestas en [1], que describimos a continuación.

En primer lugar, podemos encontrar dos conjuntos de variables innecesarias que reducen el modelo análogos a los de las Proposiciones 3.1 y 3.2. Así, observamos que, basándonos en la misma demostración que descartaba las variables del conjunto R_5 de la Proposición 3.1, tenemos la siguiente proposición:

Proposición 3.3. *Las variables del conjunto*

$$R_{5'} = \{y_{ij}^k : i,j \in K \cup K^{'}, k \in K, p \in \mathscr{P}, g_{kp} = 2, g_{ip} = g_{jp} \neq 2\}, \tag{3.46}$$

son innecesarios.

El conjunto $R_{5'}$ incluye al conjunto R_5 y lo amplía, ya que incluye variables $i,j \in K^{'}$.

Del mismo modo, la demostración de la Proposición 3.2 es aplicable a la siguiente proposición:

Proposición 3.4. *Las variables del conjunto*

$$R_{6'} = \{y_{ij}^k : i,j \in K \cup K^{'}, k \in K, p \in \mathscr{P}, g_{kp} + g_{ip} = 1 \ o \ g_{kp} + g_{jp} = 1\}. \tag{3.47}$$

son innecesarias.

Centrémonos ahora en los conjuntos $S_i,\ i \in K^{'}$ y veamos con mayor profundidad cuándo son necesarios. En la formulación del Modelo Reducido, la restricción (3.35) nos indica que, si el genotipo k pertenece a los subconjuntos $S_k, S_{k'}$, entonces estamos usando el haplotipo $h_{k'}$ para explicarlo, luego existe el conjunto $S_{k'}$ en la solución y la variable $x_{k'}$ debe valer 1. Sin embargo, se da también la desigualdad contraria, es decir, el único caso en el que utilizamos el genotipo $h_{k'}$ (y, por tanto, $x_{k'} = 1$) se da cuando hay dos subconjuntos cuyo menor genotipo es k, luego se tendrá $k \in S_k, S_{k'}$ y $y_{kk'}^k = 1$. Esto nos lleva a incluir en la formulación la restricción

$$x_{k'} \leq y_{kk'}^k \qquad \forall k \in K,$$

o, equivalentemente, sustituir la restricción (3.35) por:

$$y^k_{kk'} = x_{k'} \qquad \forall k \in K. \tag{3.48}$$

Por otra parte, si observamos los cambios en la formulación del Modelo Reducido con respecto al Modelo Básico, vemos que las restricciones (3.16) y (3.17) han sido eliminadas en el Reducido, pues no son necesarias: como los haplotipos $h_k, h_{k'}$ descritos por las variables $z_{kp}, z_{k'p}, \forall p \in \mathscr{P}$ no son necesariamente los que explican al genotipo g_k (de hecho, lo serán si $y^k_{kk'} = 1$, pero no lo serán en otro caso), podemos eliminar las restricciones de la formulación. Pero siguiendo este mismo razonamiento, podemos eliminar también la restricción (3.18), que, al igual que las anteriores, sirve para obligar a que $z_{kp} \oplus z_{k'p} = g_{kp}, \forall p \in \mathscr{P}$. Eliminamos, por tanto, la restricción (3.36) de la formulación del Modelo Reducido, reduciendo el número de restricciones necesarias.

Por último, volvamos a analizar los cambios realizados en la formulación del Modelo Reducido, en particular los referentes a la sustitución de las restricciones (3.19)-(3.22) por las (3.37)-(3.44). Si atendemos al valor de g_{kp}, que puede ser 0, 1 o 2, veremos que la restricción (3.19) del Modelo Básico es sustituida por la (3.37) en el Reducido; la (3.20) se sustituye por la (3.38); y las restricciones (3.21) y (3.22) se han sustituido por las 6 restricciones (3.39)-(3.44).

Entonces, para la restricción (3.19), que impone g_{kp} y nos interesa cuando el genotipo g_k pertenece a los conjuntos S_i, S_j e $y^k_{ij} = 1$, puede desglosarse en tres restricciones si atendemos al valor de g_{ip}. En un primer caso, cuando $g_{ip} = 1$, sabemos que los genotipos g_i, g_k son incompatibles, y por tanto es imposible que y^k_{ij} valga 1: esto es exactamente lo que nos indica el conjunto $R_{6'}$ de la Proposición 3.4; si $g_{ip} = 2$, como el haplotipo h_i que describe a g_i va a describir también a g_k, queremos $z_{ip} = 0$: esta condición nos la asegura la restricción (3.37) del Modelo Reducido; y finalmente, si $g_{ip} = 0$ con $y^k_{ij} = 1$, también necesitamos $z_{ip} = 0$. Así pues, vamos a incluir en la formulación la restricción:

$$z_{ip} \leq 1 - \sum_{j: y^k_{ij} \in \hat{\Upsilon}, j \geq i} y^k_{ij} - \sum_{j: y^k_{ij} \in \hat{\Upsilon}, j < i} y^k_{ji} \qquad \forall p \in \mathscr{P}, \forall k \in K, \; \forall i : y^k_{ij} \in \hat{\Upsilon}, g_{kp} = 0, g_{ip} = 0. \tag{3.49}$$

Desglosando de manera análoga la restricción (3.20) para el caso $g_{kp} = 1$ en 3 restricciones atendiendo al valor de g_{ip}, tenemos que el conjunto $R_{6'}$ de la Proposición 3.4 nos indica que, si $g_{ip} = 0$, g_i y g_k no son compatibles y $y^k_{ij} = 0$; la restricción (3.38) nos sirve para el caso $g_{ip} = 2$; y para el caso $g_{ip} = 1$, necesitaremos incluir la restricción:

$$z_{ip} \geq \sum_{j: y^k_{ij} \in \hat{\Upsilon}, j \geq i} y^k_{ij} + \sum_{j: y^k_{ij} \in \hat{\Upsilon}, j < i} y^k_{ji} \qquad \forall p \in \mathscr{P}, \forall k \in K, \; \forall i : y^k_{ij} \in \hat{\Upsilon}, g_{kp} = 1, g_{ip} = 1. \tag{3.50}$$

Finalmente, veamos el desglose de las restricciones (3.21)-(3.22), en las que se impone $g_{kp} = 2$, atendiendo a los valores de g_{ip} y g_{jp}. Si tenemos $g_{ip} = g_{jp} = 0$ o $g_{ip} = g_{jp} = 1$, es claro que g_k no estará explicado por h_i y h_j, pues esto es precisamente lo que indica el conjunto de variables innecesarias dado por la Proposición 3.1, así como el conjunto $R_{5'}$;

si $g_{ip} = 2$ y $g_{jp} = 0$ (o al revés), las restricciones que cubrirán estos casos son la (3.39) y la (3.40); si $g_{ip} = 2$ y $g_{jp} = 1$ (o al revés), nos servirán las restricciones (3.41) y (3.42); y por último, si $g_{ip} = g_{jp} = 2$, serán necesarias las restricciones (3.43) y (3.44) del Modelo Reducido. Así pues, vemos que con las restricciones (3.39)-(3.44) hemos cubierto todos los casos posibles cuando $g_{kp} = 2$, y en este caso no necesitamos incluir más restricciones en el modelo.

La formulación completa del Modelo Reducido con todos los cambios que acabamos de introducir se puede encontrar en la Figura 3.7.

Tras explicar el Modelo Reducido y las mejoras añadidas, retomamos el ejemplo de la Figura 3.6 de la página 38, y resolvemos el problema dado por el conjunto de genotipos $\mathfrak{G}$, esta vez utilizando el Modelo Reducido con las mejoras que acabamos de introducir[2]. Vamos a explicar brevemente las diferencias entre ambos modelos, y cómo actúan los conjuntos R_n, $n \in \{1, 2, 3_a, 3_b, 4, 5, 6\}$.

En primer lugar, vemos que en este modelo los haplotipos h_i, $h_{i'}$ no siempre resuelven al genotipo g_i: por ejemplo, $g_5 \neq h_5 \oplus h_{5'}$. Esto sucede porque hemos eliminado las restricciones (3.16)-(3.18) del Modelo Básico. Sin embargo, para las variables $y^k_{ij} = 1$, h_i y h_j sí explican a g_k: por ejemplo, para el genotipo 4, vemos que $y^4_{1'4} = 1$, y también podemos comprobar que $h_{1'p} \oplus h_{4p} = g_{1p}\ \forall p \in \{1, \ldots, 6\}$. Esto se cumple debido a la presencia de las restricciones (3.37)-(3.44), (3.49) y (3.50). Del mismo modo que en la resolución mediante el Modelo Básico, vemos que funcionan las restricciones que asignan a cada genotipo un par de haplotipos que lo explican, que si un haplotipo h_i explica a un genotipo, entonces $x_i = 1$, y que el haplotipo $x_{i'}$ solo puede utilizarse si previamente se ha usado el x_i, y en ese caso $y^i_{ii'} = 1$ (restricciones (3.33), (3.34), (3.32) y (3.48), respectivamente).

Pasamos ahora a ilustrar las variables que son declaradas como innecesarias, y fijadas a 0, por los conjuntos R_n, $n \in \{1, 2, 3_a, 3_b, 4\}$. En primer lugar, los conjuntos R_1-R_4 declaran innecesarias variables y^k_{ij} con (i) $i \geq j$, (ii) $k < i$, (iii) $k' < j$, (iv) $k = i$ y $j \neq k'$, (v) $k' = j$ y $k \neq i$, y (vi) $j = i'$ y $k \neq i$, eliminando así variables como por ejemplo (i) $y^k_{31}\ \forall k$, (ii) $y^k_{45'}$ para $k \in \{1, 2, 3\}$, (iii) $y^4_{34'}$, (iv) $y^2_{2j}\ \forall j \neq 2'$, (v) $y^3_{i3'}\ \forall i \neq 3$ y (vi) $y^k_{44'}\ \forall k \neq 4$. Estos conjuntos se describen como innecesarios en base a las condiciones impuestas de forma teórica sobre los conjuntos S_i en los que se basa el modelo, y obligan a que los S_i realmente se nombren a partir del menor genotipo que contienen, lo que no sucedía en el Modelo Básico. Así, variables como $y^1_{14'}$ que valían 1 en el Modelo Básico son fijadas a 0 en este modelo.

Veamos ahora cómo fijan variables y^k_{ij} a 0 los conjuntos R_5, R_6, $R_{5'}$ y $R_{6'}$, utilizando para ello el ejemplo de la Figura 3.8. El genotipo 1 sabemos que viene descrito por los haplotipos h_1, $h_{1'}$ en cualquier solución, por lo que veamos los conjuntos que pueden describir a g_2 y g_3. Para g_2, tenemos que pueden valer 1 las variables $\{y^2_{12}, y^2_{1'2}, y^2_{22'}\}$, y para g_3, las variables $\{y^3_{12}, y^3_{12'}, y^3_{13}, y^3_{1'2}, y^3_{1'2'}, y^3_{1'3}, y^3_{23}, y^3_{2'3}, y^3_{33'}\}$ (habiendo suprimido de estos conjuntos las variables que pertenecen a R_1-R_4). No obstante, si nos fijamos en los valores de los SNP de los tres primeros genotipos, vemos que en el SNP 2, $g_{12} = 1$ y $g_{22} = g_{32} = 0$. Esto ya nos dice que son incompatibles, y si atendemos al conjunto R_6,

[2]El código que hemos implementado en Xpress del Modelo Reducido con los cambios propuestos en este apartado se puede encontrar en el Anexo 1.

$$\text{mín} \quad \sum_{i \in K \cup K'} x_i \tag{3.31}$$

$$s.a \quad x_{i'} \leq x_i \qquad \forall i \in K \tag{3.32}$$

$$\sum_{i,j:y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k \geq 1 \qquad \forall k \in K \tag{3.33}$$

$$\sum_{i,j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k + \sum_{i,j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \leq x_i \qquad \forall k \in K, \forall i : y_{ij}^k \in \hat{\Upsilon} \tag{3.34}$$

$$y_{kk'}^k = x_{k'} \qquad \forall k \in K \tag{3.48}$$

$$z_{ip} \leq 1 - \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k - \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i : y_{ij}^k \in \hat{\Upsilon}, g_{kp} = 0, g_{ip} = 2 \tag{3.37}$$

$$z_{ip} \leq 1 - \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k - \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i : y_{ij}^k \in \hat{\Upsilon}, g_{kp} = 0, g_{ip} = 0 \tag{3.49}$$

$$z_{ip} \geq \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k + \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i : y_{ij}^k \in \hat{\Upsilon}, g_{kp} = 1, g_{ip} = 2 \tag{3.38}$$

$$z_{ip} \geq \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k + \sum_{j:y_{ij}^k \in \hat{\Upsilon}, j < i} y_{ji}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i : y_{ij}^k \in \hat{\Upsilon}, g_{kp} = 1, g_{ip} = 1 \tag{3.50}$$

$$z_{ip} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 0, g_{kp} = 2 \tag{3.39}$$

$$z_{ip} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 0, g_{jp} = 2, g_{kp} = 2 \tag{3.40}$$

$$z_{ip} \leq 1 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 1, g_{kp} = 2 \tag{3.41}$$

$$z_{ip} \leq 1 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 1, g_{jp} = 2, g_{kp} = 2 \tag{3.42}$$

$$z_{ip} + z_{jp} \geq y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2 \tag{3.43}$$

$$z_{ip} + z_{jp} \leq 2 - y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall k \in K$$

$$\forall i, j : y_{ij}^k \in Y \backslash R, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2 \tag{3.44}$$

$$x_i, z_{ip}, y_{ij}^k \in \{0, 1\}. \tag{3.45}$$

Figura 3.7: Modelo Reducido.

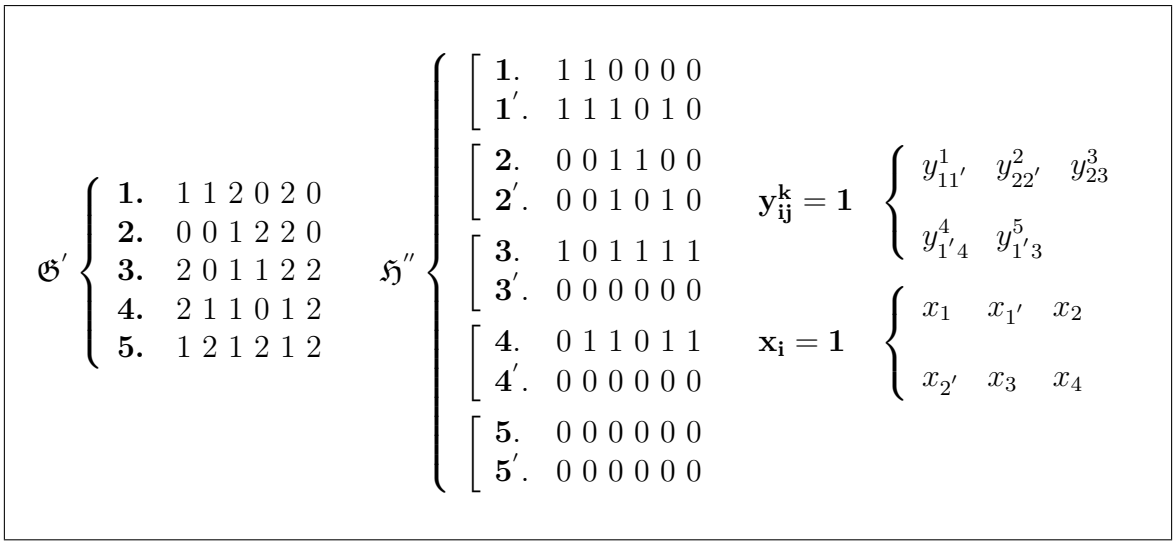


Figura 3.8: Ejemplo $\mathfrak{G}'$ y soluciones $\mathfrak{H}'' = \{h_i\}$, $i \in K \cup K'$, $\mathbf{y_{ij}^k}$ y $\mathbf{x_i}$ dadas por el programa Xpress introduciendo el código del Modelo Reducido y las variaciones propuestas en este capítulo.

vemos que $y^2_{12} = y^3_{13} = 0$, mientras que atendiendo al conjunto $R_{6'}$, $y^2_{1'2} = y^3_{1'3} = y^3_{13'} = 0$. Habiendo descartado ya dos de las tres variables que podían utilizarse para el genotipo 2, es claro que $y^2_{22'} = 1$ en la solución óptima, como en efecto sucede en la solución que tenemos en la Figura 3.8.

Por otra parte, si atendemos al SNP 6, vemos que $g_{16} = g_{26} = 0$ y $g_{36} = 2$, luego el genotipo 3 no podrá ser explicado por los haplotipos 1 y 2, es decir, $y^3_{12} = 0$, de acuerdo con R_5. Además, $R_{5'}$ nos garantiza también que no podremos combinar dos haplotipos de entre $h_1, h_{1'}, h_2$ y $h_{2'}$ para explicar a g_3, es decir, $y^3_{1'2} = y^3_{1'2'} = y^3_{12'} = 0$. De este modo, el conjunto inicial de variables que podían ser 1 pasa de tener 9 variables a tener 3, $y^3_{23}, y^3_{2'3}$ y $y^3_{33'}$. En la Figura 3.8 vemos que una de estas variables, en particular y^3_{23}, vale 1 en la solución óptima del problema.

Habiendo visto cómo actúan los conjuntos R_n, $n \in \{1, 2, 3_a, 3_b, 4, 5, 5', 6, 6'\}$ en un ejemplo concreto, podemos intuir que su inclusión en el problema es clave en la reducción de su tamaño, permitiendo al Modelo Reducido resolver instancias de mucho mayor tamaño que el Modelo Básico, y en un tiempo menor, tal y como podremos apreciar de forma más concreta en un apartado posterior.

3.2.6. Desigualdades válidas que refuerzan el Modelo Reducido

En esta sección vamos a estudiar algunas restricciones adicionales que podemos incluir para reforzar el Modelo Reducido, pues son desigualdades válidas, es decir, son satisfechas por todos los puntos factibles del problema. Todas ellas se pueden encontrar en [1].

Proposición 3.5. *Las desigualdades*

$$\sum_{k\in K: y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k \leq x_i \qquad \forall i,j : \exists y_{ij}^k \in \hat{\Upsilon}, \tag{3.51}$$

$$\sum_{k\in K: y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k \leq x_j \qquad \forall i,j : \exists y_{ij}^k \in \hat{\Upsilon} \tag{3.52}$$

son válidas para el Modelo Reducido.

Demostración.
Veamos la prueba para el conjunto de restricciones (3.51), ya que la demostración para el conjunto (3.52) es análoga. Si $x_i = 0$, entonces el haplotipo h_i no explica a ningún genotipo, luego $y_{ij}^k = 0$ para cualquier $j \in K \cup K^{'}$, $k \in K$; si $x_i = 1$, entonces la desigualdad es trivialmente válida, ya que $\sum_{k\in K} y_{ij}^k \in \{0,1\}$ porque un genotipo g_k únicamente pertenece a dos conjuntos S_i, S_j. □

Proposición 3.6. *La desigualdad*

$$z_{ip} \geq \sum_{k\in K: g_{kp}=1, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k \qquad \forall p \in \mathscr{P}, \forall i,j : \exists y_{ij}^k \in \hat{\Upsilon} \tag{3.53}$$

es válida para el Modelo Reducido.

Demostración.
Si $\sum_{k\in K: g_{kp}=1, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 0$, entonces la desigualdad es trivialmente válida; si $\sum_{k\in K: g_{kp}=1, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 1$, existe $g_k \in S_i$ con $g_{kp} = 1$, luego x_{ip} también tendrá que valer 1, y la desigualdad sigue siendo válida. □

Proposición 3.7. *La desigualdad*

$$z_{ip} + \sum_{k\in K: g_{kp}=0, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k \leq x_i \qquad \forall p \in \mathscr{P}, \forall i,j : \exists y_{ij}^k \in \hat{\Upsilon} \tag{3.54}$$

es válida para el Modelo Reducido.

Demostración.
Si $x_i = 1$ y $\sum_{k\in K: g_{kp}=0, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 1$, entonces existe un genotipo $g_k \in S_i$ con $g_{kp} = 0$, luego z_{ip} tendrá que valer 0 y la desigualdad se cumple; si $x_i = 1$ y $\sum_{k\in K: g_{kp}=0, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 0$, entonces la desigualdad es trivialmente válida; y si $x_i = 0$, entonces claramente $\sum_{k\in K: g_{kp}=0, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 0$ y, como el haplotipo h_i no se emplea en la solución, podemos fijar $z_{ip} = 0\ \forall p \in \mathscr{P}$, por lo que la desigualdad también es válida. □

Proposición 3.8. *Las desigualdades*

$$z_{ip} + z_{jp} \geq \sum_{k\in K: g_{kp}=2, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k + 2 \sum_{k\in K: g_{kp}=1, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k,$$

$$\forall p \in \mathscr{P}, \forall i,j : \exists y_{ij}^k \in \hat{\Upsilon}, \tag{3.55}$$

$$z_{ip} + z_{jp} \leq x_i + x_j - \sum_{k \in K: g_{kp}=2, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k - 2 \sum_{k \in K: g_{kp}=0, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k,$$
$$\forall p \in \mathscr{P}, \forall i, j : \exists y_{ij}^k \in \hat{\Upsilon}, \qquad (3.56)$$

son válidas para el Modelo Reducido.

Demostración.
Realizamos la prueba para las restricciones (3.55), ya que la demostración de las (3.56) sigue un razonamiento análogo. Si $z_{ip} + z_{jp} = 0$, entonces para cualquier genotipo $g_k \in S_i, S_j$, $g_{kp} = 0$, luego las variables y_{ij}^k, $k \in K$, $g_{kp} \in \{1, 2\}$ valen 0 y la desigualdad es válida; si $z_{ip} + z_{jp} = 1$, entonces cualquier genotipo $g_k \in S_i, S_j$ tendrá $g_{kp} = z_{ip} \oplus z_{jp} = 2$, luego $\sum_{k \in K: g_{kp}=1, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 0$ y la desigualdad será válida; y por último, si $z_{ip} + z_{jp} = 2$, entonces para $g_k \in S_i, S_j$, $g_{kp} = z_{ip} \oplus z_{jp} = 1$ y $\sum_{k \in K: g_{kp}=2, y_{ij}^k \in \hat{\Upsilon}} y_{ij}^k = 0$, luego la desigualdad vuelve a ser válida en este caso. □

Ahora vamos a introducir otra clase de desigualdades, utilizando para ello el concepto de genotipos incompatibles, es decir, aquellos para los que no existe ningún haplotipo que pueda explicarlos a ambos. Esta condición se da cuando, para dos genotipos g_k, g_m, existe un SNP p tal que $g_{kp} + g_{mp} = 1$. En este caso, es claro que si un haplotipo h_i explica a g_k (i.e., $\exists j : y_{ij}^k = 1$), no podrá explicar a g_m e y_{ij}^m valdrá 0 $\forall j \in K \cup K'$. Así, si tenemos un conjunto de genotipos incompatibles dos a dos g_{k_1}, ... ,g_{k_n}, obtendremos que $\sum_{k_t: t \in 1,\ldots,n} \sum_{j \in K \cup K'} y_{ij}^{k_t} \in \{0, 1\}$, $\forall i \in K \cup K'$.

Definimos el grafo $CG = (V, E)$ con V el conjunto de genotipos $\mathfrak{G}$, y $E = \{(g_i, g_j) : g_i$ y g_j son incompatibles$\}$.

Proposición 3.9. *Sea C un subgrafo completo de CG. Entonces la desigualdad*

$$\sum_{k \in C} \sum_{j: y_{ij}^k \in \hat{\Upsilon}, i \neq j} y_{ij}^k \leq x_i \qquad \forall i : y_{ij}^k \in \hat{\Upsilon}, \forall C \subset CG \qquad (3.57)$$

es válida para el Modelo Reducido.

Demostración.
Si $\sum_{k \in C} \sum_{j: y_{ij}^k \in \hat{\Upsilon}, i \neq j} y_{ij}^k = 0$, entonces la desigualdad es trivialmente válida; si $\sum_{k \in C} \sum_{j: y_{ij}^k \in \hat{\Upsilon}, i \neq j} y_{ij}^k = 1$, entonces el genotipo g_k se explica por el haplotipo h_i, luego $x_i = 1$ y la desigualdad es válida de nuevo. □

Definamos ahora $\mathfrak{G}_p^t = \{g_k \in \mathfrak{G} : g_{kp} = t\}$, para $p \in \mathscr{P}$, $t \in \{0, 1\}$, y $E_p^t = \{(g_i, g_j) \in \mathfrak{G}_p^t : \exists q \in \mathscr{P}, q \neq p, g_{ip} + g_{jq} = 1\}$, para $p \in \mathscr{P}$, $t \in \{0, 1\}$. Consideremos los grafos $CG_p^t = (\mathfrak{G}_p^t, E_p^t)$ para $t \in \{0, 1\}$, contenidos en CG, y los subgrafos completos $C^t \subset CG_p^t$, $t \in \{0, 1\}$. Entonces tenemos la siguiente proposición:

Proposición 3.10. *Las desigualdades*

$$z_{ip} \geq \sum_{g \in C^1} \sum_{j: y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k + \sum_{g \in C^1} \sum_{j: y_{ji}^k \in \hat{\Upsilon}, j < i} y_{ji}^k,$$
$$\forall p \in \mathscr{P}, \forall i : y_{ij}^k \in \hat{\Upsilon}, \forall C^1 \subset CC_p^1, \qquad (3.58)$$

		MODELO BÁSICO			MODELO REDUCIDO		
SNP	GEN.	T(s)	NODOS	RESUELTOS	T(s)	NODOS	RESUELTOS
15	10	146,92	4019,8	5/5	0,2	1	5/5
15	13	1178,7	7335	4/5	0,2	1	5/5

Tabla 3.4: Tabla comparativa entre el Modelo Básico y el Modelo Reducido.

$$z_{ip} \leq x_i - \sum_{g \in C^0} \sum_{j: y_{ij}^k \in \hat{\Upsilon}, j \geq i} y_{ij}^k - \sum_{g \in C^0} \sum_{j: y_{ji}^k \in \hat{\Upsilon}, j < i} y_{ji}^k,$$

$$\forall p \in \mathscr{P}, \forall i : y_{ij}^k \in \hat{\Upsilon}, \forall C^0 \subset CC_p^0 \qquad (3.59)$$

son válidas para el Modelo Reducido.

Demostración.
Probaremos la validez del conjunto de restricciones (3.58), ya que la del conjunto (3.59) se demuestra por analogía. Si $z_{ip} = 1$, entonces la restricción es trivialmente válida; si $z_{ip} = 0$, entonces como $g_{kp} = 1 \ \forall g_k \in C^1$, ningún genotipo g_k perteneciente a C^1 puede ser explicado por el haplotipo h_i, luego la desigualdad también es válida. □

3.2.7. Implementación de los Modelos Básico, Reducido y Reducido con algunas desigualdades válidas en Xpress y Estudio Comparativo

Para comprender mejor los modelos Básico y Reducido propuestos en [1] y poder aplicarlos a instancias concretas, hemos implementado en el programa de optimización Xpress las formulaciones del Modelo Básico, del Modelo Reducido añadiendo nuestras aportaciones, y de este último incluyendo los conjuntos de desigualdades válidas dados por las Proposiciones 3.5-3.8. Realizamos en este apartado una comparativa entre los Modelos Básico y Reducido, y entre el Modelo Reducido sin las desigualdades válidas anteriores y con ellas.

Para comparar el Modelo Básico con el Reducido, hemos introducido instancias con fragmentos de 25 SNP para conjuntos de 10 y 13 genotipos. Hemos resuelto cinco instancias de cada tipo, y en la Tabla 3.4 podemos observar el tiempo medio de resolución de los problemas para los que obtuvimos solución, así como el número de nodos del árbol de ramificación. Para instancias de 10 genotipos y 13 SNP, tenemos que el Modelo Reducido resuelve todas las instancias en 0,2 segundos, y que no tiene que emplear para ello el algoritmo de ramificación, ya que encuentra la solución en el primer nodo del árbol de ramificación. Sin embargo, el tiempo medio de ejecución para una instancia con el Modelo Básico es de 146 segundos, aprox. dos minutos y medio, mientras que la media de nodos explorados es de más de cuatro mil. Para las instancias con 10 SNP y 13 fragmentos de genotipos, vemos que el tiempo de ejecución y los nodos explorados no varían en el Modelo Reducido, ya que todas las instancias siguen resolviéndose en 0,2 segundos sin ramificar. No obstante, estas medias varían considerablemente en el Modelo Básico, en el que la media de tiempo utilizado en la resolución es de 1178,7 segundos para las cuatro instancias que se resolvieron, y la quinta no fue resuelta por el programa en más de 5000 segundos de ejecución. Con estos datos, comprobamos que las restricciones de conjuntos

		MODELO REDUCIDO			MOD. REDUCIDO + DESIG.		
SNP	GEN.	T(s)	NODOS	RESUELTOS	T(s)	NODOS	RESUELTOS
25	25	1,7	8,6	5/5	12,98	9,2	5/5
25	50	21,48	22	5/5	118,14	11,4	5/5
50	25	0,96	3,6	5/5	42,3	6	5/5
50	75	56,8	9	5/5	1556,97	2	3/5
75	50	4	1	5/5	1508,88	8,75	4/5

Tabla 3.5: Tabla comparativa entre el Modelo Reducido con y sin desigualdades válidas que lo refuerzan.

de variables de decisión y_{ij}^k son muy efectivas a la hora de reducir el tamaño del modelo, ya que al fijar todas las variables de los conjuntos R_i reducimos notablemente el número de nodos explorados por el algoritmo de ramificación, lo que se traduce en una reducción del tiempo de resolución de los ejemplos. El Modelo Básico no puede resolver en un tiempo inferior a 5000 segundos la mayor parte de instancias de 25 genotipos y 25 SNP en adelante, muchas de las cuales son resueltas por el Reducido en unos segundos, por lo que se ha excluido este modelo en la siguiente comparativa.

El tamaño de los ejemplos elaborados para comparar el Modelo Reducido con las restricciones válidas que lo refuerzan y sin ellas es muy superior al de los ejemplos anteriores, llegando a resolver instancias de 75 genotipos o 75 SNP. Para comparar estos dos modelos, hemos implementado en Xpress cinco instancias de cada uno de los tipos genotipos y SNP que se muestran en la Tabla 3.5, que, al igual que la Tabla 3.4, muestra los tiempos medios de resolución en segundos y el número medio de nodos del árbol de ramificación para los problemas que han sido resueltos, tomando como tiempo límite para ello 5000 segundos. Los ejemplos con 25 genotipos son los que menos tiempo de resolución requieren en los dos modelos, ya que el Modelo Reducido resuelve estos problemas en un tiempo medio inferior a dos segundos, mientras que el Modelo Reducido con las desigualdades válidas los resuelve, tanto para 25 SNP como para 50, en un tiempo medio inferior a un minuto. Para instancias con 50 genotipos, el tiempo medio de resolución mediante el Modelo Reducido no se incrementa en exceso, y notamos que para las instancias con 75 SNP el tiempo medio es de 4 segundos, ya que al ampliar el número de SNP aparecen más conjuntos de variables de decisión innecesarias debidos a la incompatibilidad entre genotipos (los conjuntos vistos en las Proposiciones 3.1,3.2,3.3 y 3.4). Sin embargo, al aumentar el número de SNP a 75, también aumenta el número de restricciones dadas por las Proposiciones 3.5-3.8, lo que provoca que el tiempo de resolución para estas instancias aumente en el Modelo Reducido con desigualdades con respecto de las que constan de 25 genotipos y 25 SNP, haciendo imposible su resolución en una de las instancias en menos de 5000 segundos. Finalmente, tenemos un último caso de estudio con cinco instancias de 75 genotipos y 50 SNP, las más costosas de resolver para ambos modelos. En este caso comprobamos que, mientras que el Modelo Reducido sin restricciones adicionales resuelve todas las instancias en un tiempo medio de aproximadamente un minuto, al incluir las desigualdades en el modelo el tiempo se incrementa notablemente, ya que dos de las instancias no son resueltas, y el tiempo medio de las tres restantes es de más de 1500 segundos. Comparando estos modelos para instancias de distintos tamaños, hemos comprobado que los mejores tiempos de resolución se obtienen en todos los casos con el Modelo Reducido. Esto puede deberse a que, si bien las desigualdades válidas refuerzan el problema, su inclusión en la formulación hace que aumente significativamente el número de restricciones del mismo. Con respecto al número

MODELO REDUCIDO				
SNP	GEN.	T(s)	NODOS	RESUELTOS
75	75	36,78	3,4	5/5
100	100	117,46	6	5/5

Tabla 3.6: Tabla con algunos resultados obtenidos para el Modelo Básico.

de nodos explorados durante el proceso de ramificación, la media de nodos es, en todos los casos y para los dos modelos, inferior a 25, y no hay variaciones significativas entre un modelo y otro, aunque sí se aprecia que en los casos con mayor número de genotipos que de SNP (es decir, las instancias con 50 genotipos y 25 SNP y las de 75 genotipos y 50 SNP), las restricciones adicionales ayudan a reducir el número de nodos explorados.

Para finalizar, y viendo los resultados obtenidos para el Modelo Reducido con los ejemplos anteriores, hemos incluido en Xpress instancias de 75 genotipos y 75 SNP, así como de 100 genotipos y 100 SNP, para comprobar si el Modelo Reducido podía resolverlas en un tiempo inferior a 5000 segundos. Los tiempos medios de resolución y el número medio de nodos explorados, para cinco instancias de cada tipo, se pueden observar en la Tabla 3.6. Queremos resaltar que el modelo es capaz de resolver todas las instancias, y que el tiempo medio que emplea para ello es de algo más de medio minuto para las primeras instancias, y de alrededor de dos minutos para las instancias de mayor tamaño. Este modelo proporciona, en base a lo expuesto, mejores resultados que los modelos RTIP y SC vistos en las secciones anteriores.

Capítulo 4

El Problema del Subgrafo Arcoíris Mínimo y su relación con el problema de Haplotipaje de la Parsimonia Pura

4.1. Introducción

En este capítulo, vamos a estudiar el problema del Subgrafo Arcoíris Mínimo (PSAM) y su relación con el problema de Haplotipado de la Parsimonia Pura (HPP). Así, veremos que el problema HPP se puede transformar en un problema de grafos, el PSAM. Posteriormente, presentaremos un algoritmo de aproximación en tiempo polinómico para el PSAM con un radio de aproximación de $\frac{5}{6}\Delta$, con $\Delta(G) \geq 2$ el grado máximo del grafo. Todo lo expuesto en esta sección se puede encontrar en [7].

4.2. Transformación del problema HPP en un problema de grafos

Recordemos que, para el problema HPP, dado un conjunto $\mathfrak{G}$ de p genotipos $g_1, g_2, \ldots, g_p$ correspondientes a p individuos de una población, nuestro objetivo es encontrar un conjunto $\mathfrak{H}$ de haplotipos de cardinal mínimo que *explique* $\mathfrak{G}$, es decir, tales que para cada $g \in \mathfrak{G}$ existan $h_1, h_2 \in \mathfrak{H}$ con $g = h_1 \oplus h_2$.

Introducimos ahora el siguiente

Problema del Subgrafo Arcoíris Mínimo (PSAM). *Dado un grafo G cuyas aristas están coloreadas con p colores, encuentra un subgrafo F de G de orden mínimo y con p aristas tales que cada color aparezca exactamente una vez.*

Veamos cómo podemos construir el grafo G a partir de un conjunto de genotipos:

Dado un conjunto $\mathfrak{G}$ de genotipos $g_1, g_2, \ldots, g_p$, usaremos p colores $1, 2, \ldots, p$. Para

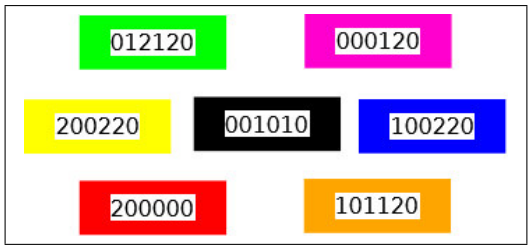


Figura 4.1: Conjunto $\mathfrak{G}$ de genotipos.

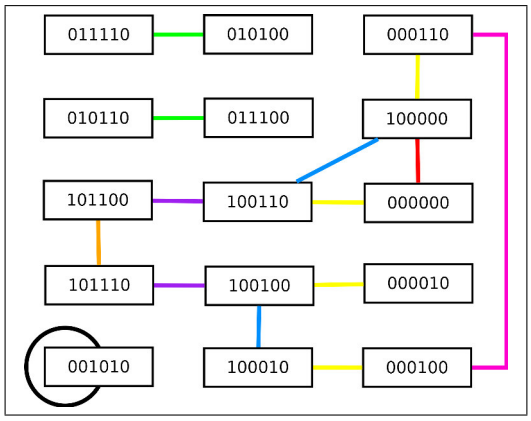


Figura 4.2: Grafo G_b obtenido a partir del conjunto $\mathfrak{G}$ de la Figura 4.1.

cada haplotipo que pueda ser utilizado para explicar un genotipo de $\mathfrak{G}$, introducimos un vértice. Si dos haplotipos $h^{'}$ y $h^{''}$ explican un genotipo g_i ($g_i = h^{'} \oplus h^{''}$), entonces se añadirá la arista $(h^{'}, h^{''}$ de color i a G. Si un genotipo se explica por dos haplotipos idénticos ($g_i = h \oplus h$), entonces al vértice correspondiente se le añade una arista llamada *bucle*, es decir, una arista de la forma (h, h). Como ejemplo, en la Figura 4.1 podemos ver un conjunto de genotipos $\mathfrak{G}$ que dan lugar al grafo G_b de la Figura 4.2. Así, el genotipo de color amarillo de la Figura 4.1, como tiene 3 posiciones ambiguas (dadas por los doses), puede explicarse por $2^3 = 8$ haplotipos, que son los nodos sobre los que inciden las aristas amarillas de la Figura 4.2. En cambio, el genotipo de color negro no tiene posiciones ambiguas, por lo que está formado por dos haplotipos iguales y da lugar a un bucle en la Figura 4.2.

Este proceso da lugar a la construcción de un grafo G_b cuyas aristas están coloreadas por p colores. Como un haplotipo se puede usar como máximo una vez en un par de haplotipos que expliquen un genotipo, obtenemos una *arista-coloración propia*, es decir, ningún vértice posee dos aristas incidentes del mismo color. Además, cualquier conjunto $\mathfrak{H}$ de haplotipos que explique $\mathfrak{G}$ se corresponderá con un *subgrafo arcoíris* (i.e., un subgrafo que contiene los p colores) H de G_b. Es claro que encontrar un conjunto $\mathfrak{H}$ mínimo de haplotipos que expliquen $\mathfrak{G}$ es equivalente a encontrar un subgrafo F de G_b (con G_b obtenido a partir de $\mathfrak{G}$) que tenga p aristas, cada una de un color, y que sea mínimo (es decir, lo que llamamos un *subgrafo arcoíris mínimo*). En tal caso, el conjunto H vendrá dado por los nodos de F, y para cada genotipo g_i existirá una única arista de color i en F que incidirá en los nodos $h^{'}$ y $h^{''}$, luego tendremos g_i explicado. Así, podemos transformar el problema HPP en un problema de optimización en grafos, el PSAM.

Una vez establecida la relación entre el problema de Haplotipado de la Parsimonia

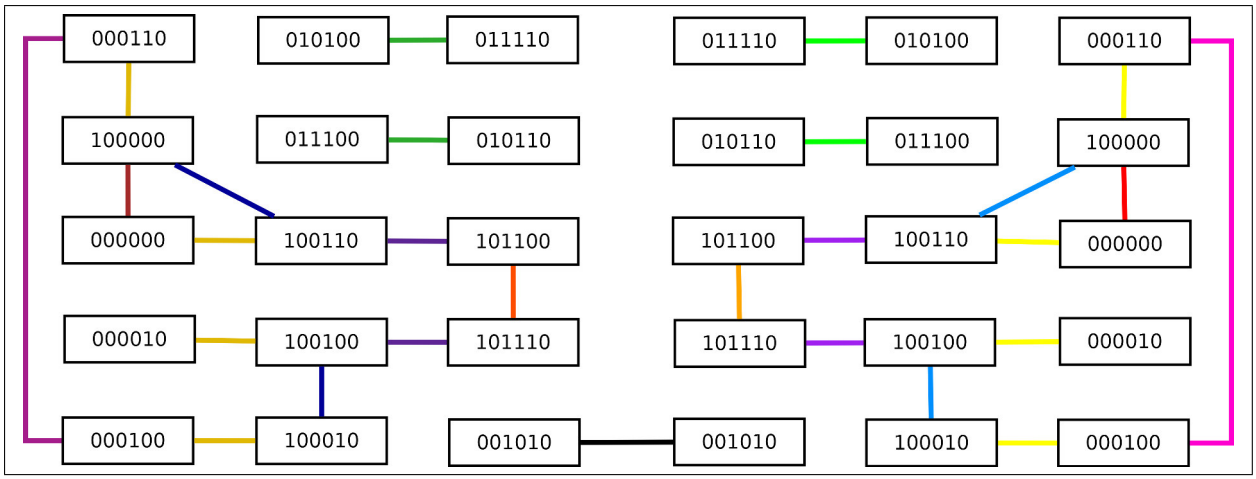


Figura 4.3: Grafo G^* simple obtenido a partir del grafo G_b de la Figura 4.2.

Pura y el problema del Subgrafo Arcoíris Mínimo, vamos a centrarnos en el estudio de este último para grafos simples, es decir, sin bucles. Para ello, vamos a considerar primero una serie de resultados que nos permitirán asumir que el grafo dado en el PSAM es simple.

Que un genotipo no tenga posiciones ambiguas es equivalente a que solo pueda ser resuelto por dos haplotipos idénticos. Por tanto, en el grafo correspondiente, existirá un bucle dado por este haplotipo. Claramente, este bucle será la única arista del grafo que tenga el color dado por el genotipo, por lo que el vértice estará contenido en cualquier subgrafo arcoíris.

Vamos a construir un grafo simple G^* a partir de G_b. Sea $V(G^*) = V(G_1) \cup V(G_2)$ el conjunto de nodos de $V(G^*)$, donde G_1 y G_2 son dos copias de G_b. Sean $V(G_1) = \{v_1, v_2, \ldots, v_n\}$ y $V(G_2) = \{w_1, w_2, \ldots, w_n\}$ y supongamos que $\{v_1, v_2, \ldots, v_t\}$ y $\{w_1, w_2, \ldots, w_t\}$ son los $2t$ vértices en los que incide un bucle. Eliminemos ahora estos $2t$ bucles y añadamos las aristas $(v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)$. Cada arista (v_i, w_i), $1 \leq i \leq t$, se coloreará con el mismo color de los dos bucles (v_i, v_i) y (w_i, w_i) eliminados, y si una arista e de G_1 recibe el color i, entonces la misma arista e de G_2 recibirá el color $p+i$. Por tanto, las aristas de G^* estarán coloreadas con $2p - t$ colores y $\Delta(G^*) \leq \Delta(G_b)$, con $\Delta(G^*)$ y $\Delta(G_b)$ los grados máximos de G^* y G_b, respectivamente.

En la Figura 4.3 podemos observar el grafo simple G^* obtenido a partir de G_b. La arista bucle de color negro incidente en el nodo (001010) se ha eliminado en este grafo, siendo sustituida por una arista $(001010, 001010)$ de color negro que conecta los dos nodos iguales de G_1 (copia de G_b de la derecha) y G_2 (copia de G_b de la izquierda). Además, el conjunto de aristas de color verde claro en G_b tienen el mismo color en G_2, mientras que en G_1 están coloreadas de otro color, en este caso verde oscuro.

Supongamos ahora que aplicamos cualquier algoritmo de aproximación en el grafo G^* y obtenemos un subgrafo H^* de G^*. Sea $V(H_i) = V(H^*) \cap V(G_i)$ para $i = 1, 2$ y supongamos, sin pérdida de generalidad, que $\min(|V(H_1)|, |V(H_2)|) = |V(H_1)|$. Sabiendo que, para F^* subgrafo arcoíris mínimo de G^* y F subgrafo arcoíris mínimo de G_b, $|V(F^*)| = 2|V(F)|$, y se obtiene:

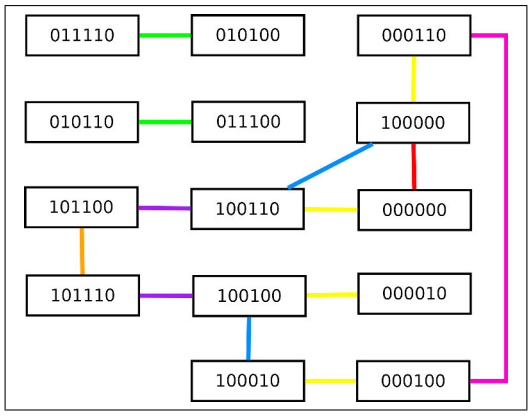


Figura 4.4: Grafo $G \subset G_b$ simple.

$$\frac{|V(H^*)|}{|V(F^*)|} = \frac{|V(H_1)| + |V(H_2)|}{2|V(F)|} \geq \frac{\text{mín}(|V(H_1)|, |V(H_2)|)}{|V(F)|} = \frac{|V(H_1)|}{|V(F)|}.$$

Por lo tanto, $\frac{|V(H^*)|}{|V(F^*)|} \geq \frac{|V(H_1)|}{|V(F)|}$. Esto prueba que una garantía de comportamiento de $1+\epsilon$ para el PSAM para grafos simples implica una garantía de comportamiento de $1+\epsilon$ para grafos con bucles. Así, si encontramos un algoritmo de aproximación con un radio de aproximación para H^* simple (respecto del subgrafo mínimo F^*), este mismo radio también será de aproximación para H_1 (respecto de F). Por lo tanto, de ahora en adelante podemos asumir que G_b no tiene bucles. En la Figura 4.4 observamos el grafo simple $G \subset G_b$ que utilizaremos en los ejemplos siguientes.

4.3. Algoritmo de aproximación del PSAM con un radio de aproximación de $\Delta(G)$

Veamos ahora un algoritmo de aproximación en tiempo polinómico para el problema del Subgrafo Arcoíris Mínimo con un radio de aproximación de $\Delta(G)$.

Dado un grafo G simple, escogiendo de forma arbitraria una arista de cada color podemos encontrar en tiempo polinómico un subgrafo arcoíris H' de G de orden $|V(H')| \leq 2p$. Por otra parte, si H' tiene p aristas, la suma de grados de sus nodos es $2p$, y si el grado máximo de G es $\Delta(G)$, entonces es claro que el número de nodos de H' será mayor o igual que $\frac{2p}{\Delta(G)}$. Así, tenemos cotas inferior y superior para el número de vértices de H':

$$\frac{2p}{\Delta(G)} \leq |V(H')| \leq 2p. \tag{4.1}$$

Entonces, para un subgrafo arcoíris mínimo F tendremos, haciendo uso de las cotas anteriores:

$$\frac{|V(H')|}{|V(F)|} \leq \frac{2p}{|V(F)|} \leq \frac{2p}{\frac{2p}{\Delta(G)}} = \Delta(G). \tag{4.2}$$

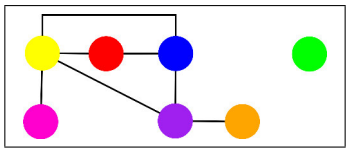


Figura 4.5: Grafo G' obtenido aplicando el Paso 1 del Algoritmo 4.1.

De esta forma, hemos hallado un algoritmo de aproximación al PSAM en tiempo polinómico con un radio de aproximación de $\Delta(G)$.

4.4. Algoritmo de aproximación del PSAM con un radio de aproximación de $\frac{5}{6}\Delta(G)$

En esta sección, vamos a demostrar el siguiente

Teorema 4.1. *El PSAM puede aproximarse en tiempo polinómico con un radio de aproximación de $\frac{5}{6}\Delta(G)$ para grafos de grado máximo $\Delta(G) \geq 2$.*

Para el caso en el que $\Delta(G) = 1$, G está inducido por un emparejamiento, y por tanto cualquier subgrafo arcoíris alcanzará la cota superior $2p$ y tendrá orden mínimo. Podemos asumir entonces que $\Delta(G) \geq 2$.

A continuación introducimos un algoritmo para encontrar en tiempo polinómico un subgrafo arcoíris H de G no necesariamente mínimo que utilizaremos para probar el Teorema 4.1. Observemos, además, que dos aristas adyacentes de diferentes colores en G inducen un subgrafo de tres vértices, mientras que dos aristas de diferentes colores de un emparejamiento inducen un subgrafo de cuatro vértices.

Algoritmo 4.1. *Construcción de un sugrafo H arcoíris de G.*

1. *Construye un grafo G' con $V(G') = \{v_1, v_2, \ldots, v_p\}$ (v_i se corresponde con el color i) y $v_iv_j \in E(G')$ si existen dos aristas adyacentes $e, f \in E(G)$ con $c(e) = i$ y $c(f) = j$, donde $c(i)$ denota el color de la arista i.*

2. *Ahora encuentra un emparejamiento máximo M de tamaño m_M en G'. El algoritmo de construcción de un emparejamiento máximo en un grafo se detalla en la sección 1.2 del Capítulo 1.*

3. *Por último, construye un grafo H con $V(H) \subseteq V(G)$ de la forma siguiente: para cada arista del emparejamiento M elige dos aristas adyacentes en G con estos dos colores, formando un camino de grado 3 que llamaremos P_3. Para cada vértice de $V(G')$ que no esté en M elige una arista en G con este color. Así, hemos elegido una arista de cada color, por lo que obtenemos un grafo H con $|E(H)| = p$. H es, por tanto, un subgrafo arcoíris de G.*

Vamos a ilustrar el Algoritmo 4.1 con un ejemplo. Aplicando el Paso 1 del Algoritmo 4.1 al grafo G de la Figura 4.4, obtenemos el grafo G', que podemos observar en la Figura

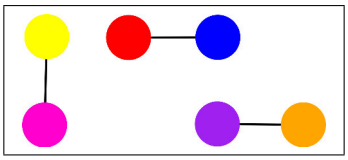


Figura 4.6: Emparejamiento máximo M obtenido aplicando el Paso 2 del Algoritmo 4.1.

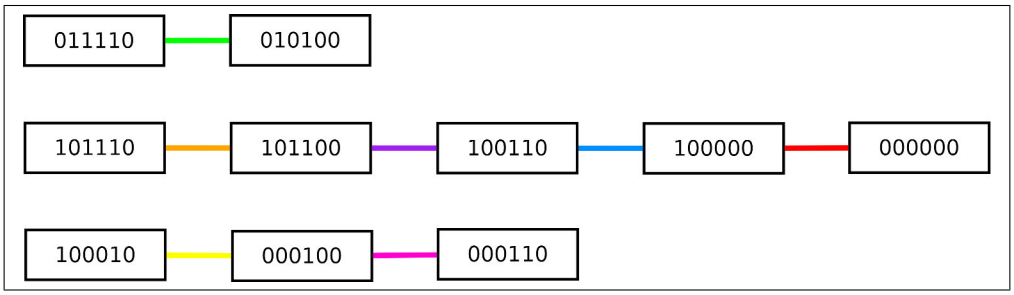


Figura 4.7: Subgrafo arcoíris $H \subseteq G$ obtenido aplicando el Paso 3 del Algoritmo 4.1.

4.5. Así, en el grafo de la Figura 4.5, los nodos representan los haplotipos, y vemos que exite una arista que conecta los nodos amarillo y rosa porque en la Figura 4.4 existen dos aristas amarilla y rosa que inciden en un mismo nodo (por ejemplo el (000100)). En cambio, no existen aristas rosa y azul que incidan en un mismo nodo, por lo que en G' no existe la arista que conecta los nodos rosa y azul. El Paso 2 consiste en encontrar un emparejamiento máximo $M \subseteq G'$, que en nuestro ejemplo es único y lo podemos observar en la Figura 4.6. A partir del grafo M, aplicando el Paso 3 del Algoritmo 4.1, obtenemos un subgrafo arcoíris $H \subseteq G$ que podemos observar en la Figura 4.7. En ella podemos ver que nodos como el verde, que no pertenecen al emparejamiento M, dan lugar a componentes conexas de dos nodos, mientras que nodos adyacentes en M como el amarillo y el rosa dan lugar a componentes conexas con al menos tres nodos. Además, puede suceder que al escoger los nodos en el grafo inicial, haya nodos que formen parte de más de uno de estos caminos P_3, como en este ejemplo sucede con el nodo (100110), dando lugar a componentes conexas de más de tres nodos, ya que los nodos se pueden escoger más de una vez en el Paso 3 del Algoritmo 4.1.

Es importante resaltar que, mediante el Algoritmo 4.1, el grafo H tiene una particularidad que utilizaremos más adelante en la demostración del Teorema 4.1: H cuenta con el mayor número de caminos P_3 arista-disjuntos dos a dos que se pueden formar con un conjunto de p aristas cada una de un color en G. En efecto, cada arista del emparejamiento M constituye un camino P_3 en H, y estos caminos son arista-disjuntos, ya que no se repiten aristas de un mismo color. Por ser M máximo, tenemos que el número de caminos P_3 también lo es.

Ahora que tenemos un subgrafo arcoíris $H \subseteq G$, queremos ver que $\frac{|V(H)|}{|V(F)|} \leq \frac{5}{6}\Delta(G)$ para F subgrafo arcoíris mínimo, es decir, queremos probar que hemos hallado un algoritmo que en un tiempo polinómico resuelve el problema del Subgrafo Arcoíris Mínimo con un radio de aproximación de $\frac{5}{6}\Delta(G)$, demostrando así el Teorema 4.1. Damos primero una proposición y un teorema que nos servirán en la demostración del Teorema 4.1:

Proposición 4.1. *$|V(H)| \leq 2p - m_M$, con m_M el tamaño de M.*

Demostración. Para cada arista $e \in M$, de las cuales hay m_M, tenemos tres nodos en H; para cada nodo $v' \in G \backslash M$, de los cuales hay $p - 2m_M$, tenemos dos nodos en H. Sabiendo que algunos de estos nodos pueden coincidir en H, tenemos:

$$|V(H)| \leq 3m_M + 2(p - 2m_M) = 2p - m_M.$$

□

Ahora probaremos el siguiente

Teorema 4.2. *Sea G un grafo conexo de tamaño m. Entonces G contiene $\lfloor \frac{m}{2} \rfloor$ caminos P_3 arista-disjuntos dos a dos de orden tres*[1]*.*

Demostración. Realizaremos la prueba por inducción en m. Si $1 \leq m \leq 2$, entonces la afirmación es trivialmente verdadera. Supongamos que G tiene tamaño $m+1$ para algún $m \geq 2$. Sea $e \in E(G)$ una arista cualquiera de G.

- Supongamos que e es una arista puente, es decir, tal que $G - e = (V, E - \{e\})$ no es conexo, y sean u_1, u_2 los extremos de e. Sean G_1', G_2' las dos componentes conexas de $G - e$ tales que $u_i \in V(G_i')$ para $i = 1, 2$. Si G_1' o G_2' tienen tamaño par, entonces podemos hacer una partición de su conjunto de aristas en caminos arista-disjuntos de orden tres. Por el contrario, si ambos tienen tamaño impar, entonces $G_1' + e = (V_1', E_1' \cup \{e\})$ tiene tamaño par y se puede hacer una partición de su conjunto de aristas en caminos P_3 arista-disjuntos de orden tres. En ambos casos aplicamos la hipótesis de inducción en las aristas restantes.

- Asumamos ahora que $G - e$ es conexo. Si e es adyacente a alguna arista $f \in E(G)$ tal que $G - \{e, f\}$ es conexo, entonces las aristas e, f inducen un camino de orden tres y la afirmación se sigue haciendo uso de la hipótesis de inducción. Podemos asumir, por tanto, que cualquier arista f adyacente a e es una arista puente de $G - e$. Sean v_1, v_2 los dos extremos de la arista f y supongamos que e es incidente con v_1. Denotemos por G_1, G_2 los dos subgrafos de $G - \{e, f\}$ tales que $v_i \in V(G_i)$ para $i = 1, 2$. Si G_1 o G_2 tiene tamaño impar, entonces la afirmación sigue por inducción. Si los dos tienen tamaño par, entonces $G_1 + e$ y $G_2 + f$ tienen tamaño par los dos, y de nuevo la afirmación se sigue por inducción.

□

Ahora procedemos con la demostración del Teorema 4.1. Dado un grafo conexo G de tamaño m, sea $\beta_3(G)$ el número máximo de caminos P_3 arista-disjuntos dos a dos contenidos en G. Por el Teorema 4.2, sabemos que $\beta_3(G) = \lfloor \frac{m}{2} \rfloor$.

Consideremos ahora las componentes conexas $F_1, F_2, \ldots, F_k$ del subgrafo arcoíris mínimo F de G. Sean $n_i = |V(F_i)|$ y $m_i = |E(F_i)|$ el número de vértices y aristas de cada componente conexa F_i, respectivamente. Por el Teorema 4.2, tenemos que $\beta_3(F_i) = \lfloor \frac{mi}{2} \rfloor$

[1]Para el lector interesado, un algoritmo de construcción de dichos caminos puede hallarse en el Anexo 2.

es el máximo número de caminos P_3 arista-disjuntos dos a dos para cada componente conexa F_i de F, $i = 1, \ldots, k_f$.

Sea $d_i = 2m_i - n_i$. Entonces

$$m' := \sum_{i=1}^{k} d_i = \sum_{i=1}^{k}(2m_i - n_i) = 2\sum_{i=1}^{k} m_i - \sum_{i=1}^{k} n_i = 2p - |V(F)|,$$

luego

$$|V(F)| = 2p - m'. \tag{4.3}$$

Además, hemos visto anteriormente que H tiene el número máximo de caminos P_3 arista-disjuntos dos a dos para un subgrafo arcoíris de G, luego usando que $\lfloor \frac{E(H_i)}{2} \rfloor = \beta_3(H_i)$ el máximo número de caminos P_3 arista-disjuntos dos a dos de H_i componente conexa de H, $i = 1, \ldots, k_h$, podemos concluir que

$$m_M = \sum_{i=1}^{k_h} \beta_3(H_i) = \sum_{i=1}^{k_h} \left\lfloor \frac{E(H_i)}{2} \right\rfloor \geq \sum_{i=1}^{k_f} \left\lfloor \frac{E(F_i)}{2} \right\rfloor =: \sum_{i=1}^{k_f} \beta_3(F_i). \tag{4.4}$$

Teniendo en cuenta las expresiones (4.3) y (4.4) y utilizando $\beta_3(F_i) = c_i \cdot d_i$ para $i = 1, \ldots, k$, se tiene:

$$\begin{aligned} |V(H)| &\leq 2p - m_M \leq 2p - \sum_{i=1}^{k} \beta_3(F_i) = 2p - \sum_{i=1}^{k} c_i \cdot d_i \leq \\ &\leq 2p - \sum_{i=1}^{k} c \cdot d_i = 2p - c \cdot m', \end{aligned} \tag{4.5}$$

con $c = \text{mín}\{c_1, c_2, \ldots, c_k\}$.

Y por la expresión (4.1) aplicada al subgrafo F, tenemos $2p - m' = |V(F)| \geq \frac{2p}{\Delta(G)}$, que implica

$$2p - m' \geq \frac{m'}{\Delta(G) - 1}, \tag{4.6}$$

ya que $(2p - m')\Delta(G) \geq 2p \Rightarrow (2p - m')(\Delta(G) - 1) \geq 2p - (2p - m') = m'$.

Corolario 4.1. *Utilizando las expresiones* (4.3), (4.5) *y* (4.6), *obtenemos*

$$\frac{|V(H)|}{|V(F)|} \leq \frac{2p - c \cdot m'}{2p - m'} = 1 + \frac{(1-c)m'}{2p - m'} = 1 + \frac{(1-c)m'}{\frac{m'}{\Delta(G)-1}} = 1 + (1-c)(\Delta(G) - 1).$$

Vamos a llevar a cabo ahora estimaciones para la constante c:

Corolario 4.2. $\frac{|V(H)|}{|V(F)|} \leq \frac{5}{6}\Delta(G)$

Demostración. Para cada componente conexa F_i, sea $|V(F_i)| + \frac{1}{c_i} \cdot \lfloor \frac{m_i}{2} \rfloor = 2m_i$. Entonces $c_i \geq \lfloor \frac{m_i}{2} \rfloor \cdot \frac{1}{2m_i} \cdot \frac{\Delta(G)}{\Delta(G)-1}$, ya que $|V(F_i)| \geq \frac{2m_i}{\Delta(G)}$. Por tanto, nos queda

$$c_i \geq \lfloor \frac{m_i}{2} \rfloor \cdot \frac{1}{2m_i} \cdot \frac{\Delta(G)}{\Delta(G)-1} \geq \frac{m_i - 1}{2} \cdot \frac{1}{2m_i} \cdot \frac{\Delta(G)}{\Delta(G)-1} \geq \frac{2}{3} \cdot \frac{1}{4} \cdot \frac{\Delta(G)}{\Delta(G)-1} \geq \frac{1}{6} \cdot \frac{\Delta(G)}{\Delta(G)-1}.$$

Sustituyendo en el Corolario 4.1, tenemos la demostración:

$$\begin{aligned} \frac{|V(H)|}{|V(F)|} &\leq 1 + (1-c)(\Delta(G)-1) \leq 1 + (1 - \frac{1}{6} \cdot \frac{\Delta(G)}{\Delta(G)-1})(\Delta(G)-1) \\ &= 1 + \Delta(G) - 1 - \frac{\Delta(G)}{6} = \frac{5}{6}\Delta(G). \end{aligned}$$

□

La prueba del Teorema 4.1 se sigue del Corolario 4.2.

Por último, veamos el valor de c para $\Delta(G) = 2$. En este caso, las componentes conexas de F pueden ser caminos o ciclos. Veamos en cada componente el valor de c_i:

1. Si F_i es un ciclo de grado par C_{2q}, con $q \geq 2$, entonces $|E(F_i)| = |V(F_i)|$, y $d_i = 2|E(F_i)| - |V(F_i)| = |E(F_i)| = 2q$. Sabiendo que $q = \lfloor \frac{2q}{2} \rfloor = \beta_3(F_i) = c_i \cdot d_i = c_i \cdot 2q$, obtenemos $c_i = \frac{1}{2}$.

2. Si F_i es un ciclo de grado impar C_{2q+1}, con $q \geq 1$, entonces $|E(F_i)| = |V(F_i)|$, y $d_i = 2|E(F_i)| - |V(F_i)| = |E(F_i)| = 2q + 1$. Sabiendo que $q = \lfloor \frac{2q+1}{2} \rfloor = \beta_3(F_i) = c_i \cdot d_i = c_i \cdot 2q + 1$, obtenemos $c_i = \frac{q}{2q+1} \geq \frac{1}{3}$.

3. Si F_i es un camino de tamaño par P_{2q}, $q \geq 2$, $d_i = 2|E(F_i)| - |V(F_i)| = 2(2q) - (2q+1) = 2q - 1$. Entonces es claro que $q = \lfloor \frac{2q}{2} \rfloor = \beta_3(F_i) = c_i \cdot d_i = c_i \cdot 2q - 1$ y $c_i = \frac{q}{2q-1} > \frac{1}{2}$.

4. Por último, si F_i es un camino de tamaño impar P_{2q+1}, $q \geq 1$, $d_i = 2|E(F_i)| - |V(F_i)| = 2(2q+1) - (2q+2) = 2q$, y se tiene $q = \lfloor \frac{2q+1}{2} \rfloor = \beta_3(F_i) = c_i \cdot d_i = c_i \cdot 2q$, con lo que $c_i = \frac{1}{2}$.

Así, tenemos que $c = c(2) = \frac{1}{3}$, el mínimo de c_i para cada F_i componente conexa de F.

Conclusiones

Finalizamos este trabajo con un breve apartado en que se destacan algunas de las conclusiones obtenidas y de los resultados que nos ha proporcionado, tanto a nivel personal como en relación a los contenidos adquiridos durante el proceso de elaboración y redacción del mismo.

La realización de este trabajo me ha servido para ampliar los conocimientos de algunas asignaturas cursadas en el grado, como *Grafos y Optimización Discreta*, y en particular el último capítulo del trabajo me ha permitido aplicar estos conocimientos a un problema de la vida real y de interés actual.

La Optimización Entera nos permite establecer una relación entre las Matemáticas y otras ciencias experimentales, en este caso la Biología y la Genética, a través de problemas que surgen en estas áreas y que motivan la elaboración de modelos matemáticos para tratar de resolverlos. La mayor parte de las veces no existe un modelo teórico que nos resuelva de forma óptima todas las instancias, ya que muchos de estos problemas son complejos, tienen varios objetivos y presentan muchas variaciones. No obstante, su estudio induce un proceso de investigación que consiste en la búsqueda de formulaciones que mejoren los resultados proporcionados por las anteriores, como en el caso del problema HPP aquí analizado.

En concreto, el estudio en profundidad del Modelo Reducido propuesto en [1], así como la implementación que he llevado a cabo en Xpress, me ha permitido dar una solución a algunos errores encontrados en su formulación, sin la cual no era posible resolver correctamente ejemplos del problema HPP.

Sería interesante para un trabajo futuro realizar un estudio comparativo más extenso de los tres modelos de [1], así como analizar qué conjuntos de desigualdades válidas propuestas son más efectivos a la hora de reducir el tiempo requerido para la resolución de distintas instancias.

Por otra parte, el trabajo ha supuesto personalmente un cambio de perspectiva con respecto a otras asignaturas del grado, al permitirme una colaboración más personalizada con un profesor, y me ha introducido en el proceso de búsqueda, análisis y sistematización de la información que requiere todo trabajo de investigación, a la vez que iniciarme en la actividad investigadora.

Anexos

Anexo 1

Formulaciones del problema HPP en Xpress

En este anexo, vamos a incluir el código que hemos implementado en el programa de optimización Xpress para resolver el problema de Haplotipaje de la Parsimonia Pura, tanto mediante el Modelo Básico, como mediante el Reducido con las mejoras añadidas en la sección 3.2.5, y también incluyendo en el Reducido el código de algunas de las desigualdades válidas dadas en el apartado 3.2.6, concretamente las de las Proposiciones 3.5-3.8.

A la hora de introducir el código en Xpress, hemos declarado las variables con el mismo nombre que adquieren a lo largo del trabajo, y hemos comentado las restricciones con el número de referencia que tienen, para que se pueda ver de manera sencilla a qué restricción corresponde cada sección del código. Asímismo, para definir las variables x_i, z_i e y_{ij}^k cuando i o j pertenecen a $K \cup K^{'}$, se utilizan los índices $l, m \in \{1, 2\}$, donde el valor 1 indica que la variable pertenece a K, y el valor 2, que pertenece a $K^{'}$. Por ejemplo, la variable dada en el código como $x(i, 1)$ representa al haplotipo h_i, mientras que la variable $x(i, 2)$ representa a $h_{i'}$.

En ambos códigos, incluimos un ejemplo sencillo dado por los genotipos $g_1 = (2222)$, $g_2 = (1202)$ y $g_3 = (0212)$.

1.1. Código del Modelo Básico

```
declarations
        n = 3   !Número de hijos a resolver
    rgi = 1..n
    rgl = 1..2
    rgj = 1..n
    rgm = 1..2
    rgk = 1..n
    rgp = 1..4  !Número de SNP
    g : array (rgk, rgp) of integer
    x : array(rgi, rgl) of mpvar
    z : array (rgi, rgl, rgp) of mpvar
    y : array (rgi, rgl, rgj, rgm, rgk) of mpvar
end-declarations

forall(i in rgi, l in rgl) x(i,l) is_binary
forall(i in rgi, l in rgl, p in rgp)  z(i,l,p) is_binary
forall(i in rgi, l in rgl, j in rgj, m in rgm, k in rgk) y(i,l,j,m,k)
   is_binary

g::[2,2,2,2,
    1,2,0,2,
    0,2,1,2]


forall(k in rgk,i in rgi,j in rgj,m in rgm,l in rgl) do
 !Para fijar i<j
 if i>j then
        y(i,l,j,m,k) = 0
 end-if
 if i=j and l>=m then
        y(i,l,j,m,k) = 0
 end-if
end-do

!(3.12)
forall (i in rgk) do
 !(2)
 x(i,2) <= x(i,1)
end-do


!(3.13)
forall(k in rgk) do
 sum(i in rgi,j in rgj,l in rgl,m in rgm) y(i,l,j,m,k) = 1
end-do

!(3.14)
forall(i in rgi,l in rgl,k in rgk) do
 sum(j in rgj,m in rgm | j>i or (j=i and m>l)) y(i,l,j,m,k) + sum(j in rgj,
    m in rgm | i>j or (i=j and l>m)) y(j,m,i,l,k) <= x(i,l)
end-do

!(3.15)
forall (k in rgk) do
```

```
 y(k,1,k,2,k) <= x(k,2)
end-do

forall (k in rgk,p in rgp) do
!(3.16)
 if(g(k,p)=0) then
  z(k,1,p) = 0
  z(k,2,p) = 0
 end-if
!(3.17)
 if(g(k,p)=1) then
  z(k,1,p) = 1
  z(k,2,p) = 1
 end-if
!(3.18)
 if(g(k,p)=2) then
  z(k,1,p) + z(k,2,p) = 1
 end-if
end-do

!(3.19)
forall(i in rgi,l in rgl, k in rgk, p in rgp  | g(k,p)=0 and i<>k) do
 z(i,l,p) <= 1 - sum (j in rgj,m in rgm | j>i or (j=i and m>l)) y(i,l,j,m,k
     ) - sum (j in rgj,m in rgm | i>j or (i=j and l>m)) y(j,m,i,l,k)
end-do

!(3.20)
forall(i in rgi,l in rgl, k in rgk,p in rgp | g(k,p)=1 and i<>k) do
 z(i,l,p) >= sum (j in rgj,m in rgm |  j>i or (j=i and m>l)) y(i,l,j,m,k) +
      sum (j in rgj,m in rgm | i>j or (i=j and l>m)) y(j,m,i,l,k)
end-do

forall(i in rgi,l in rgl,k in rgk,p in rgp,j in rgj,m in rgm | g(k,p)=2 and
     (i<j or (i=j and l<m ))) do
 !(3.21)
 y(i,l,j,m,k) <= z(i,l,p) + z(j,m,p)
 !(3.22)
 y(i,l,j,m,k) <= 2- z(i,l,p) - z(j,m,p)
end-do

!Función objetivo: (3.11)
minimize(sum (i in rgi,l in rgl) x(i,l))
```

1.2. Código del Modelo Reducido

```
declarations
        n = 3   !Número de genotipos a resolver
    rgi = 1..n
    rgl = 1..2
    rgj = 1..n
    rgm = 1..2
    rgk = 1..n
    rgp = 1..4  !Número de SNP
    g : array (rgk, rgp) of integer
    x : array(rgi, rgl) of mpvar
    z : array (rgi, rgl, rgp) of mpvar
    y : array (rgi, rgl, rgj, rgm, rgk) of mpvar
    fijamos : array (rgi, rgl, rgj, rgm, rgk) of integer
end-declarations

forall(i in rgi, l in rgl) x(i,l) is_binary     !Esta restricción se puede
   quitar porque la integridad de las variables y_ij^k es suficiente para
   garantizar la de x_i en el óptimo
forall(i in rgi, l in rgl, p in rgp)  z(i,l,p) is_binary
forall(i in rgi, l in rgl, j in rgj, m in rgm, k in rgk) y(i,l,j,m,k)
   is_binary

g::[2,2,2,2,
    1,2,0,2,
    0,2,1,2]

!Fijamos y_{11'}^1 = 1
y(1,1,1,2,1) = 1
! Eliminamos las variables pertenecientes a los conjuntos R
forall(k in rgk,i in rgi,j in rgj,m in rgm,l in rgl) do

 !R1
 if i>j then
        fijamos(i,l,j,m,k):=1
        y(i,l,j,m,k) = 0
 end-if
 if i=j and l>=m then
        fijamos(i,l,j,m,k):=1
        y(i,l,j,m,k) = 0
 end-if

 !R2
 if k<j then
        y(i,l,j,m,k) = 0
        fijamos(i,l,j,m,k):=1
 end-if
 if ((k=i and l=1) and (k<>j or m=1)) or (k=i and l=2) then
  fijamos(i,l,j,m,k):=1
  y(i,l,j,m,k) = 0
 end-if

 !R3a y R3b
 if (i=j and l=1 and m=2) and k<>i then
  fijamos(i,l,j,m,k):=1
```

```
  y(i,l,j,m,k) = 0
 end-if!)
 end-do

 !R4
 if ((k=j and m=2) and (k<>i or l=2))  then
  fijamos(i,l,j,m,k):=1
  y(i,l,j,m,k) = 0
 end-if

 !R5'
 forall(i in rgi,l in rgl, j in rgj, m in rgm, k in rgk, p in rgp) do
  if (g(k,p) = 2 and g(i,p) = 1 and g(j,p) = 1)  then
   fijamos(i,l,j,m,k):=1
   y(i,l,j,m,k) = 0
  end-if
  if (g(k,p) = 2 and g(i,p) = 0 and g(j,p) = 0)  then
   y(i,l,j,m,k) = 0
   fijamos(i,l,j,m,k):=1
  end-if
 end-do

 !R6'
 forall(i in rgi, l in rgl, j in rgj, m in rgm, k in rgk, p in rgp) do
  if (g(k,p) + g(i,p) = 1 or g(k,p) + g(j,p) = 1)  then
   y(i,l,j,m,k) = 0
   fijamos(i,l,j,m,k):=1
  end-if
 end-do

!(3.32)
forall (i in rgi) x(i,2) <= x(i,1)

!(3.33)
forall(k in rgk) do
sum(i in rgi,j in rgj,l in rgl,m in rgm) y(i,l,j,m,k) >= 1
end-do

!(3.34)
forall(i in rgi,l in rgl,k in rgk) do
 sum(j in rgj,m in rgm |fijamos(i,l,j,m,k) <> 1) y(i,l,j,m,k) + sum(j in
     rgj,m in rgm |fijamos(j,m,i,l,k) <> 1) y(j,m,i,l,k) <= x(i,l)
end-do

!(3.48)
forall (k in rgk) do
 y(k,1,k,2,k) = x(k,2)
end-do

!(3.37) y (3.49)
forall (i in rgi, l in rgl, k in rgk, p in rgp |g(k,p) = 0 and (g(i,p) = 2
    or g(i,p) = 0)) do
 z(i,l,p) <= 1 - sum(j in rgj, m in rgm | fijamos(i,l,j,m,k) <> 1) y(i,l,j,
     m,k) - sum(j in rgj, m in rgm | fijamos(j,m,i,l,k) <> 1) y(j,m,i,l,k)
end-do

!(3.38) y (3.50)
```

```
forall (i in rgi, l in rgl, k in rgk, p in rgp |g(k,p) = 1 and (g(i,p) = 2
    or g(i,p) = 1)) do
 z(i,l,p) >= sum(j in rgj, m in rgm | fijamos(i,l,j,m,k) <> 1) y(i,l,j,m,k)
     + sum(j in rgj, m in rgm | fijamos(j,m,i,l,k) <> 1) y(j,m,i,l,k)
end-do

forall (i in rgi, l in rgl, j in rgj, m in rgm, p in rgp, k in rgk | g(k,p)
    = 2 and fijamos(i,l,j,m,k) <> 1) do
 if(g(i,p) = 2 and g(j,p) = 0) then
  !(3.39)
  z(i,l,p) >= y(i,l,j,m,k)
 end-if
 if(g(i,p) = 0 and g(j,p) = 2) then
  !(3.40)
  z(j,m,p) >= y(i,l,j,m,k)
 end-if
 if(g(i,p) = 2 and g(j,p) = 1) then
  !(3.41)
  z(i,l,p) <= 1 - y(i,l,j,m,k)
 end-if
 if(g(i,p) = 1 and g(j,p) = 2) then
  !(3.42)
  z(j,m,p) <= 1 - y(i,l,j,m,k)
 end-if
 if(g(i,p) = 2 and g(j,p) = 2) then
  !(3.43) y (3.44)
  z(i,l,p) + z(j,m,p) >= y(i,l,j,m,k)
  z(i,l,p) + z(j,m,p) <= 2 - y(i,l,j,m,k)
 end-if
end-do

!Función objetivo: (3.31)
minimize(sum (i in rgi,l in rgl) x(i,l))
```

Anexo 2

Algoritmo de construcción de $\left\lfloor \frac{m}{2} \right\rfloor$ caminos P_3 arista-disjuntos dos a dos en un grafo G conexo

En la página 59, el Teorema 4.2 establece que, dado un grafo G conexo, podemos encontrar $\lfloor \frac{m}{2} \rfloor$ caminos P_3 arista-disjuntos dos a dos. Una vez que sabemos que esta partición puede hacerse para G conexo, vamos a ver un algoritmo que la realiza en tiempo polinómico. Este algoritmo se puede encontrar en [7].

Algoritmo 2.1. *Encuentra un árbol generador T de G (i.e., un árbol $T = (V, E')$ con $E' \subseteq E$).*

1. *Sea $G - E' = (V, E - E')$ y $\Delta(G - E')$ el grado máximo de $G - E'$. Si $\Delta(G - E') > 1$, elige dos aristas incidentes y forma un camino P_3 de orden tres.*
2. *Si $G - E'$ tiene una arista incidente con una hoja de T, forma con estas dos aristas un P_3.*
3. *Si $G - E'$ tiene una arista incidente con el vecino de una hoja, toma esta arista y la incidente en la hoja y forma un P_3.*
4. *Si dos hojas tienen el mismo vecino en T, forma un camino P_3 con ambas.*
5. *Si el vecino de una hoja tiene grado dos, escoge un camino P_3 empezando en esa hoja.*

Estos pasos se llevan a cabo con preferencia decreciente, es decir, un paso solo se realiza si ninguno de los anteriores es posible. El algoritmo finaliza cuando queda como máximo una arista que no pertenece a ningún P_3.

Vamos ahora a ilustrar el funcionamiento del Algoritmo 2.1 utilizando para nuestro ejemplo el grafo G de la Figura 2.1. Así, lo primero que realiza el algoritmo es construir un árbol generador $T \subseteq G$, que podemos encontrar (con los nodos y aristas en verde) en la Figura 2.2.

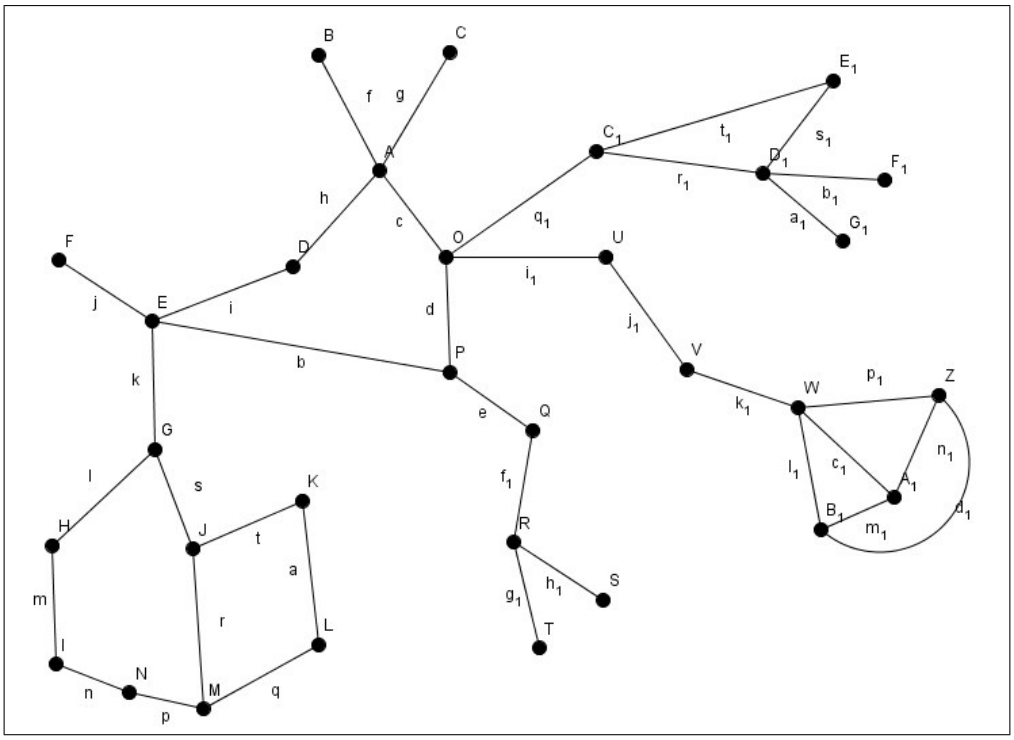


Figura 2.1: Grafo G.

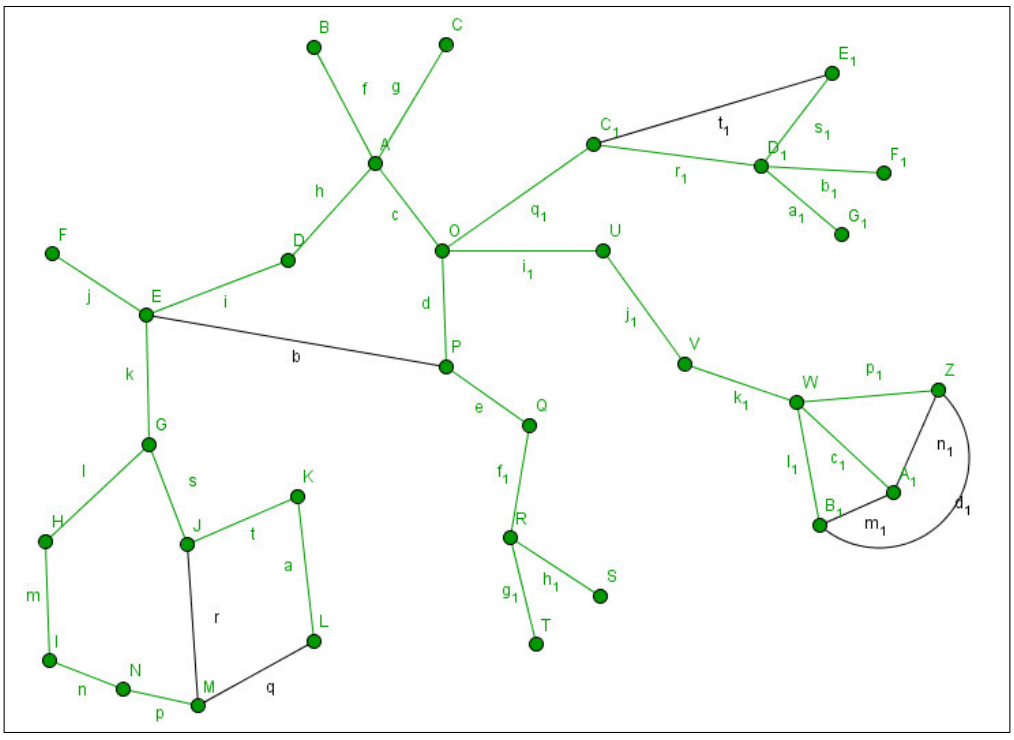


Figura 2.2: En verde, árbol generador T de G. En negro, aristas de $G - T$.

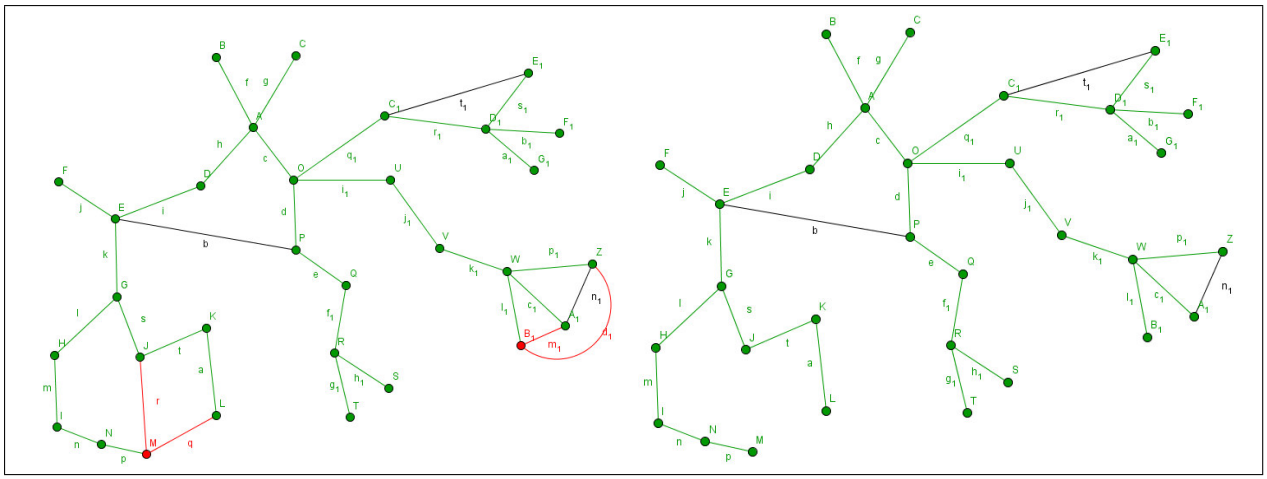


Figura 2.3: Aplicando el paso 1 del algoritmo 2.1.

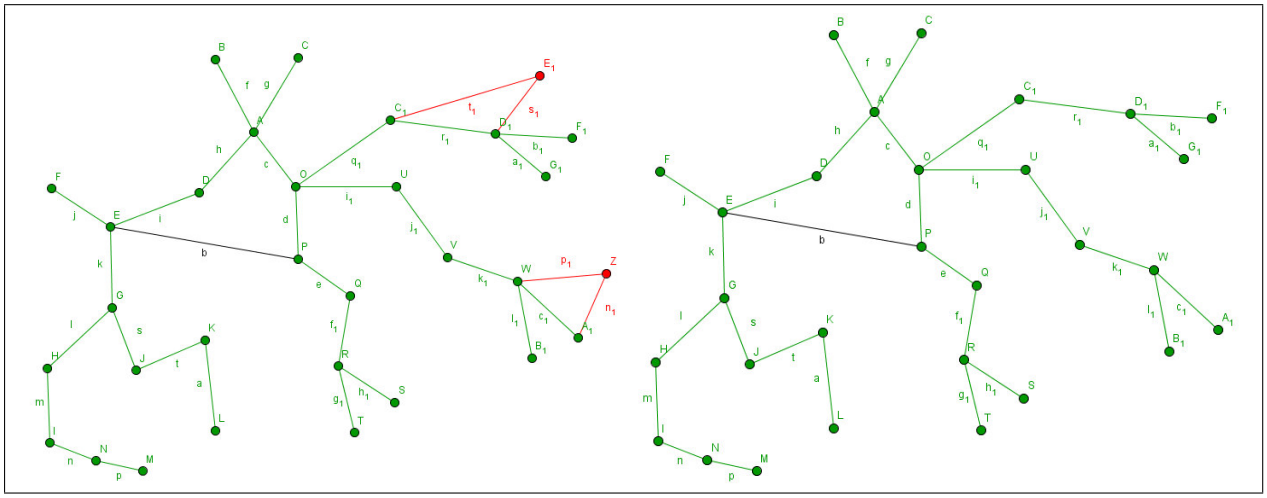


Figura 2.4: Aplicando el paso 2 del algoritmo 2.1.

Una vez tenemos T, aplicamos el Paso 1 del algoritmo tantas veces como se pueda. En nuestro ejemplo, en la Figura 2.3 podemos ver que el Paso 1 se ha aplicado dos veces, obteniendo dos caminos P_3 arista-disjuntos de grado tres (que eliminamos de nuestro grafo para seguir aplicando el algoritmo). Cuando no se puede aplicar el Paso 1 más veces, pasamos a aplicar el Paso 2: en la Figura 2.4 comprobamos cómo queda el grafo G tras aplicar el paso 2 dos veces.

Del mismo modo, cuando ya no podemos aplicar ninguno de los dos primeros pasos, aplicamos el Paso 3, y obtenemos el grafo de la Figura 2.5. Tras esto, como ya no podemos aplicar los tres primeros pasos, obtenemos la Figura 2.6 aplicando el Paso 4 cuatro veces.

Ahora aplicamos el Paso 5 del Algoritmo 2.1 sucesivas veces, proceso que se puede observar en la Figura 2.7. En este grafo comprobamos que no podemos aplicar los tres primeros pasos, por lo que aplicando el Paso 4 de nuevo una única vez nos queda solo una arista que no pertenece a ningún P_3, con lo que aquí finaliza el Algoritmo 2.1. Este paso se muestra en la Figura 2.8. El grafo G y su descomposición en caminos P_3 arista-disjuntos dos a dos obtenida tras la aplicación del Algoritmo 2.1 se puede observar en la Figura 2.9.

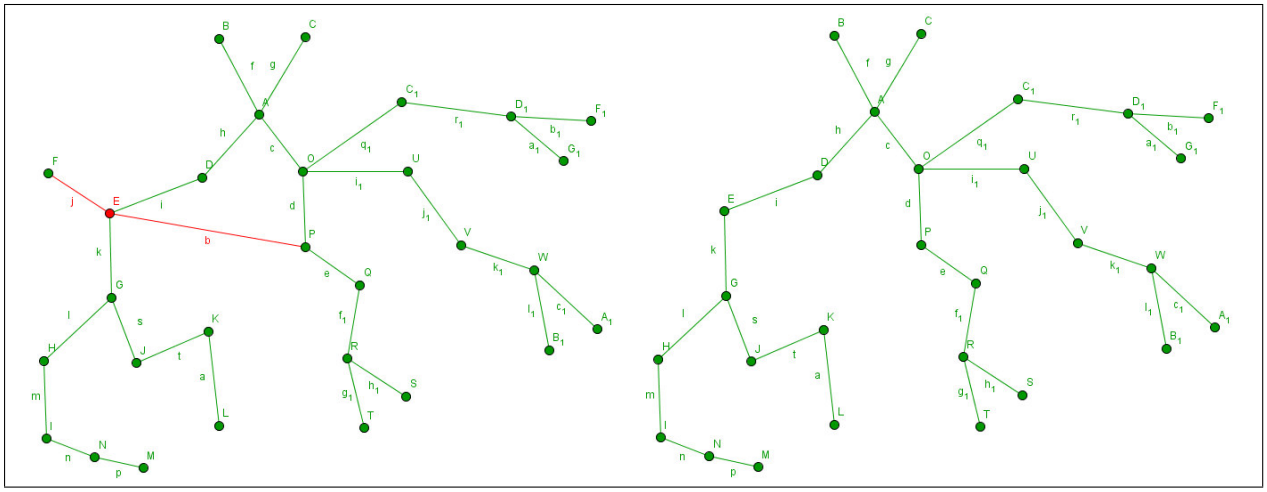


Figura 2.5: Aplicando el paso 3 del algoritmo 2.1.

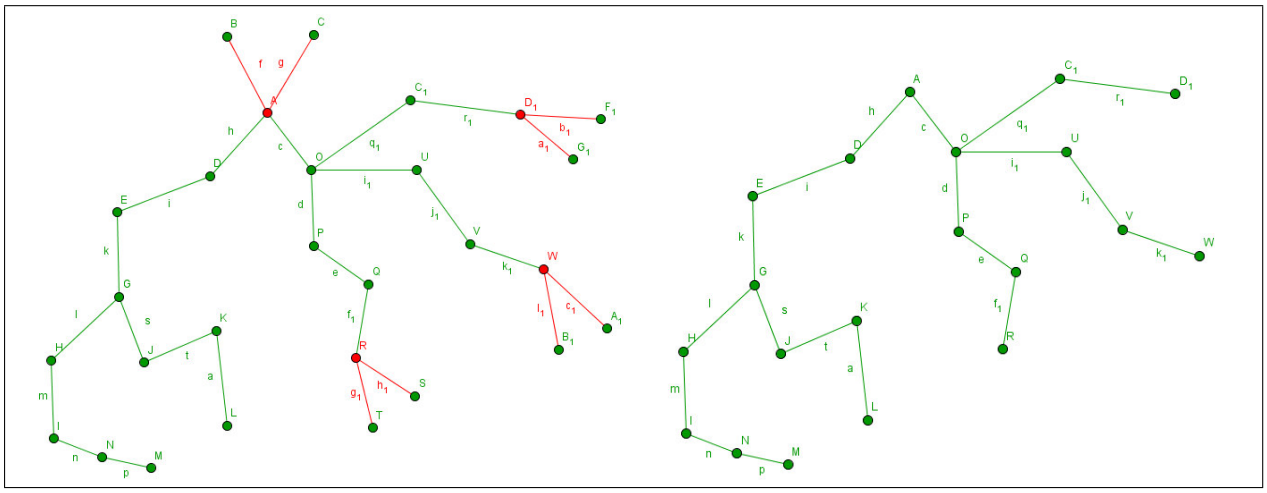


Figura 2.6: Aplicando el paso 4 del algoritmo 2.1.

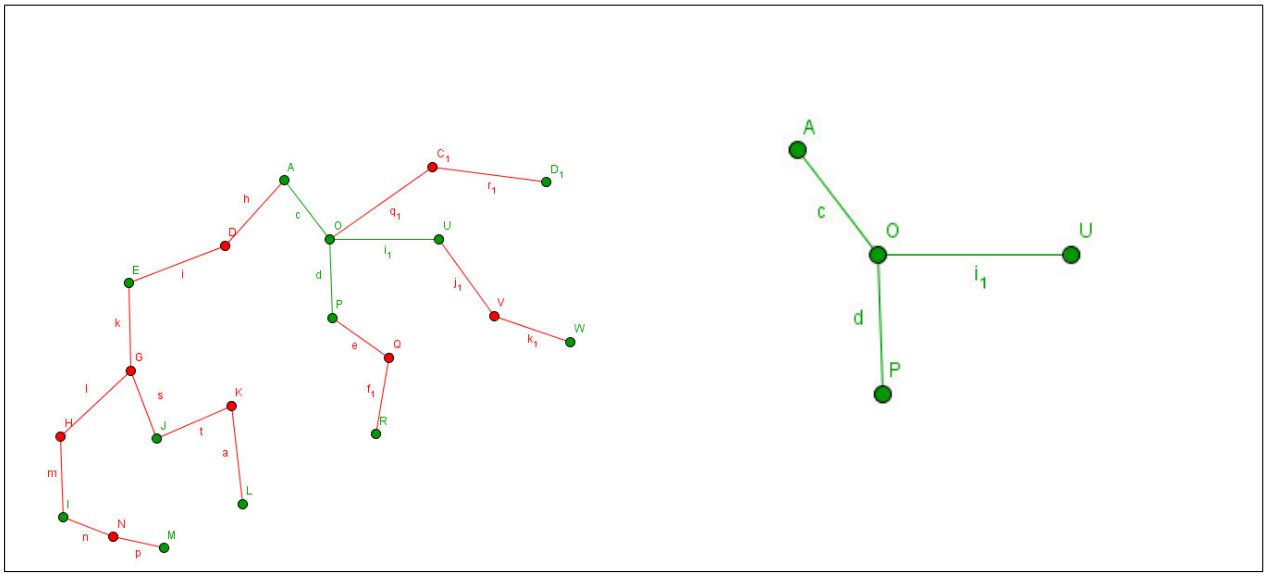


Figura 2.7: Aplicando el paso 5 del algoritmo 2.1.

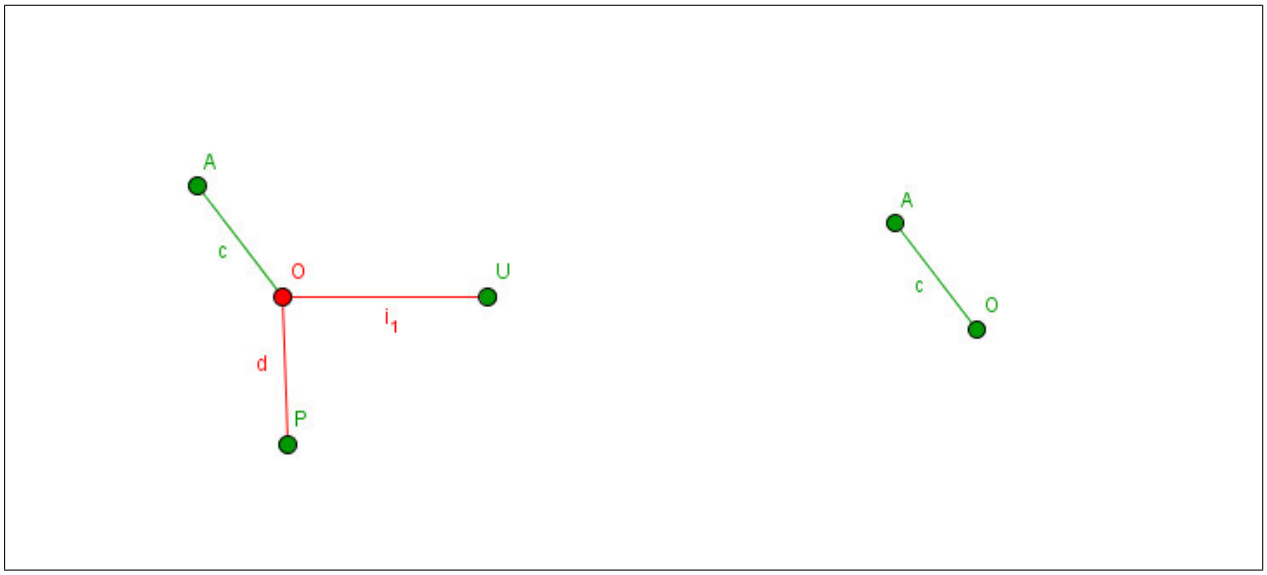


Figura 2.8: Aplicando de nuevo el paso 4 finaliza el algoritmo.

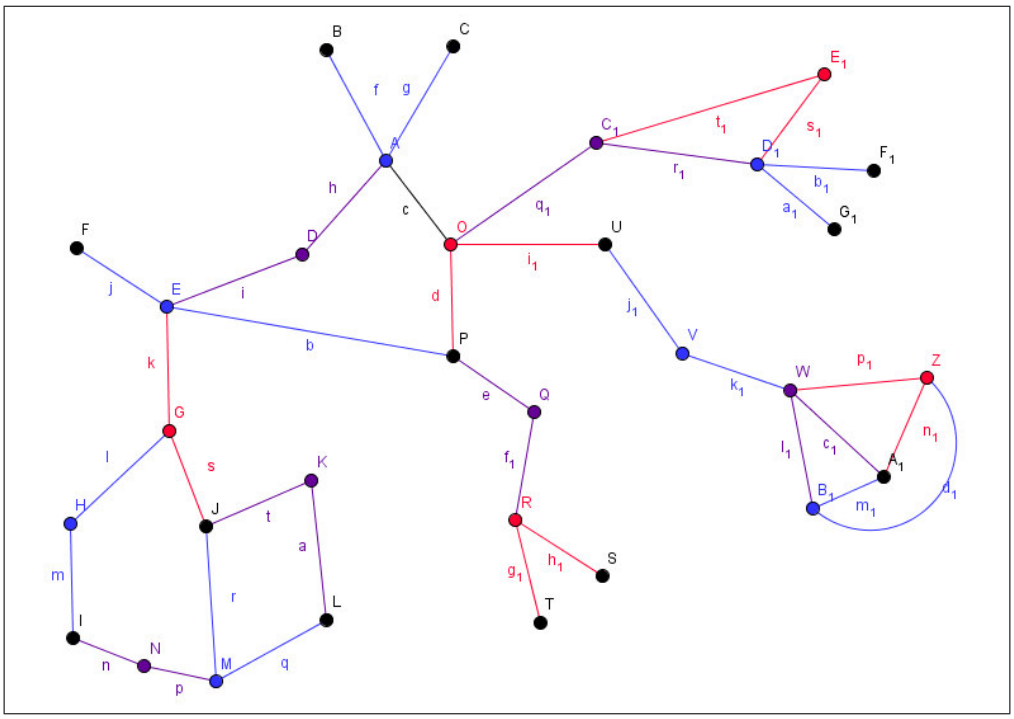


Figura 2.9: P_3 Obtenidos en el grafo G tras la aplicación del Algoritmo 2.1.

Bibliografía

[1] CATANZARO, D., A. GODI, M. LABBÉ. 2010. A Class Representative Model for Pure Parsimony Haplotyping. *INFORMS Journal on Computing*, **22**(2), pp. 195-209. ISSN 1091-9856.

[2] GUSFIELD, D. 2003. Haplotype inference by pure parsimony. En: *Proceedings of the Annual Symposium on Combinatorial Pattern Matching (CPM), Lecture Notes in Computer Science*, **2676**, pp. 144-155.

[3] JIMENO FERNÁNDEZ, A., M. BALLESTEROS VÁZQUEZ, L. UGEDO UCAR. *Nova: Biología*. Torrelaguna, Madrid, España: Grupo Santillana de Ediciones, S.A., 2000. ISBN 84-294-6597-9.

[4] LANCIA, G. Combinatorial Haplotyping Problems. En: *Pattern Recognition in Computational Molecular Biology: Techniques and Approaches.* Hoboken, NJ, USA.: John Wiley & Sons, Inc., 2015, pp. 1-28. ISBN 9781119078845.

[5] LANCIA, G., P. SERAFINI. 2009. A set covering approach wich column generation for parsimony haplotyping. *INFORMS Journal on Computing*, **21**(1), pp.151-166.

[6] MARÍN, A. 2014-2015. Apuntes de la asignatura *Grafos y Optimización Discreta*.

[7] MATOS CAMACHO, S., I. SCHIERMEYER, Z. TUZA. 2010. Approximation algorithms for the minimum rainbow subgraph problem. *Discrete Mathematics*, **310**, pp. 2666-2670.

[8] PELEGRÍN, B., L. CÁNOVAS, P. FERNÁNDEZ. Emparejamiento Máximo. En: *Algoritmos en Grafos y Redes.* Barcelona, España: Promociones y Publicaciones Universitarias, S.A., 1992, pp. 209-228. ISBN 84-477-0031-3.

[9] PIERCE, B.A. *Genética: Un enfoque conceptual.* 3ª ed. Madrid, España: Editorial Médica Panamericana, S.A., 2009. [fecha de consulta 10 julio 2016]. Disponible en:

`https://books.google.es/books?id=ALR9bgLtFhYC&pg=PA556&dq=gen%C3%A9tica+haplotipos&hl=en&sa=X&ved=0ahUKEwjs2I_siL7NAhVLVRQKHZvNBWQQ6AEINzAC#v=onepage&q&f=false`

[10] TYMOCZKO, J.L., J.M. BERG, L. STRYER. *Bioquímica: Curso Básico*. Barcelona, España: Editorial Reverté, S.A., 2014. ISBN 978-84-291-7603-2.