UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

FACULTAD DE INGENIERIA

Elaboración de un fixture para los campeonatos profesionales del fútbol uruguayo

Martín Lago

Nicolás Lantean

Santiago Rodríguez

Supervisor:
Martín Pedemonte

Ingeniería en Computación
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Abril de 2022

RESUMEN

El fútbol en Uruguay es, sin lugar a dudas, el deporte más popular, al punto de representar una parte importante de la cultura del país. No es de sorprender que el Campeonato Uruguayo, la principal competencia de fútbol del país, sirva de entretenimiento a gran parte de la población. El proceso de elaborar el fixture de un campeonato es un proceso complicado, cuyo resultado influye en ciertos factores como qué tan justo es para los equipos, qué tan atractivo es para los fanáticos o qué tan rentable es económicamente para la organización del torneo. En este proyecto se presentan distintos algoritmos que intentan optimizar estos objetivos tanto individualmente como en conjunto. Se utilizaron algoritmos de programación entera para crear algoritmos que optimizan un objetivo individual, así como algoritmos evolutivos multi-objetivo para optimizar más de un objetivo simultáneamente. Se analizaron los resultados obtenidos por los distintos algoritmos y se compararon tanto entre ellos como también con ejemplos reales, no sólo del Campeonato Uruguayo de Fútbol, si no que también con ejemplos de la Liga Uruguaya de Básquetbol. Estos resultados obtenidos demostraron la efectividad de los algoritmos para conseguir soluciones de calidad, por lo que pueden servir de guía para incentivar el uso de técnicas de optimización para la creación de fixtures en nuestro país, con el objetivo de cumplir de una mejor manera con los objetivos deseados.

Palabras claves:
Fixture, Campeonato Uruguayo de fútbol, Programación Entera, Algoritmo Evolutivo Multi-objetivo.

Tabla de contenidos

Lista de figuras viii

Lista de tablas ix

1 Introducción 1

2 Conceptos preliminares y estado del arte 7

2.1 Conceptos preliminares 7

2.2 Relevamiento del estado del arte 10

2.2.1 Relevamiento de artículos sobre fútbol 12

2.2.1.1 Scheduling the Belgian Soccer League (Goossens and Spieksma, 2009) 12

2.2.1.2 Scheduling the professional Ecuadorian football league by Integer Programming (Recalde et al., 2013) 13

2.2.1.3 Scheduling the English Football League with a Multi-objective Evolutionary Algorithm (While and Kendall, 2014) 15

2.2.1.4 Scheduling the South American Qualifiers to the 2018 FIFA World Cup by Integer Programming (Durán et al., 2017) 16

2.2.1.5 Scheduling the Australian Football League (Kyngäs et al., 2017) 17

2.2.1.6 Operations Research Transforms the Scheduling of Chilean Soccer Leagues and South American World Cup Qualifiers (Alarcón et al., 2017) 18

2.2.2 Relevamiento de artículos sobre otros deportes 19

2.2.2.1 An application of the traveling tournament problem: The Argentine volleyball league (Bonomo et al., 2012) . . . 19
2.2.2.2 Scheduling Argentina's professional basketball leagues: A variation on the Travelling Tournament Problem (Durán et al., 2019) . . . 21
2.2.3 Resumen de los artículos relevados . . . 22

3 Modelado del problema 27
3.1 Primera versión del modelo . . . 27
3.1.1 Restricciones generales . . . 28
3.1.2 Restricciones round robin . . . 28
3.1.3 Restricciones específicas . . . 28
3.1.4 Posibles funciones objetivo . . . 29
3.2 Segunda versión del modelo . . . 34
3.2.1 Restricciones generales . . . 35
3.2.2 Restricciones round robin . . . 35
3.2.3 Restricciones específicas . . . 35
3.2.4 Posibles funciones objetivo . . . 36
3.3 Tercera versión del modelo . . . 39
3.3.1 Restricciones generales . . . 40
3.3.2 Restricciones round robin . . . 40
3.3.3 Restricciones específicas . . . 41
3.3.4 Posibles funciones objetivo . . . 42

4 Algoritmo multi-objetivo para el diseño de un fixture 45
4.1 Optimización multi-objetivo . . . 45
4.2 Dominancia . . . 46
4.3 Frente de Pareto . . . 47
4.4 Hipervolumen . . . 48
4.5 Algoritmos evolutivos . . . 49
4.5.1 Algoritmo NSGA-II . . . 52
4.6 Algoritmo evolutivo multi-objetivo para la elaboración de un fixture . . . 54
4.6.1 Método del polígono . . . 55
4.6.2 Operador de mutación . . . 58

4.6.3 Función de fitness 59

5 Análisis experimental **61**
5.1 Campeonatos reales utilizados para el análisis 61
5.2 Análisis de resultados con algoritmo de programación entera . . 64
5.3 Análisis experimental del algoritmo NSGA-II 69
5.3.1 Análisis del efecto del tamaño de población 69
5.3.2 Análisis del efecto de la cantidad de iteraciones 72
5.3.3 Análisis de convergencia 73
5.3.4 Análisis de los resultados obtenidos 76

6 Conclusiones y trabajo futuro **83**

Referencias bibliográficas **87**

Apéndices **89**

1 Fixtures reales **91**

2 Fixtures obtenidos **101**

Lista de figuras

4.1 Espacio de decisión con su correspondiente espacio de objetivos. Imagen tomada de Denysiuk, 2014. 47
4.2 Hipervolumen con respecto a las soluciones no dominadas. Imagen tomada de Deb, 2001. 49
4.3 Ejemplo de cruzamiento de un punto 52
4.4 Esquema de funcionamiento del algoritmo NSGA-II. Imagen tomada de Coello et al., 2007. 53
4.5 Ejemplo utilizando el método del polígono para ocho equipos. Imagen tomada de Barone et al., 2006. 56
4.6 Fecha de los partidos para cada par de equipos utilizando el método del polígono. 57
4.7 Localías de los partidos para cada par de equipos. 57

5.1 Promedio del hipervolumen en 10 ejecuciones para diferentes tamaños de población 71
5.2 Promedio del tiempo en 10 ejecuciones para diferentes tamaños de población 71
5.3 Hipervolumen a lo largo de las generaciones 72
5.4 Tiempo a lo largo de las generaciones 73
5.5 Breaks local vs efecto arrastre 74
5.6 Breaks visitante vs efecto arrastre 75

Lista de tablas

5.1 Análisis de fixtures reales 62
5.2 Minimización de breaks 66
5.3 Minimización de efecto arrastre 66
5.4 Comparación de fixtures para el Campeonato Uruguayo de Fútbol 78
5.5 Comparación de fixtures para la Liga Uruguaya de Básquetbol del año 2020/21 78
5.6 Comparación de fixtures para la Liga Uruguaya de Básquetbol del año 2021/22 80

A1.1 Fixture del Campeonato Uruguayo de Fútbol 2020 91
A1.2 Fixture del Campeonato Uruguayo de Fútbol 2021 94
A1.3 Fixture de la Liga Uruguaya de Básquetbol 2020/21 97
A1.4 Fixture de la Liga Uruguaya de Básquetbol 2021/22 99

A2.1 Fixture del Campeonato Uruguayo obtenido con el algoritmo de Programación Entera minimizando los breaks 101
A2.2 Fixture del Campeonato Uruguayo obtenido con el algoritmo de Programación Entera minimizando el efecto arrastre . . 104
A2.3 Fixture de la Liga Uruguaya de Básquetbol 2020/21 obtenido con el algoritmo de Programación Entera minimizando los breaks 107
A2.4 Fixture de la Liga Uruguaya de Básquetbol 2020/21 obtenido con el algoritmo de Programación Entera minimizando el efecto arrastre 109
A2.5 Fixture de la Liga Uruguaya de Básquetbol 2021/22 obtenido con el algoritmo de Programación Entera minimizando los breaks 111

A2.6 Fixture de la Liga Uruguaya de Básquetbol 2021/22 obtenido con el algoritmo de Programación Entera minimizando el efecto arrastre 113
A2.7 Fixture que mejor desempeño obtuvo en breaks de los obtenidos usando el algoritmo evolutivo multi-objetivo para el Campeonato Uruguayo de Fútbol 115
A2.8 Fixture que mejor desempeño obtuvo en efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para el Campeonato Uruguayo de Fútbol 118
A2.9 Fixture que obtuvo un desempeño balanceado en breaks y efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para el Campeonato Uruguayo de Fútbol . 121
A2.10 Fixture que mejor desempeño obtuvo en breaks de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2020/21 124
A2.11 Fixture que mejor desempeño obtuvo en efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2020/21 126
A2.12 Fixture que obtuvo un desempeño balanceado en breaks y efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2020/21 128
A2.13 Fixture que mejor desempeño obtuvo en breaks de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2021/22 130
A2.14 Fixture que mejor desempeño obtuvo en efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2021/22 132
A2.15 Fixture que obtuvo un desempeño balanceado en breaks y efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2021/22 134

Capítulo 1

Introducción

El fútbol en nuestro país es el deporte por excelencia, por encima de cualquier otro, atrayendo a gran parte de la población, tanto a observarlo como a practicarlo. Por otra parte, el uso de los avances tecnológicos no está exento de formar parte y facilitar el desarrollo del mundo del deporte, es por ello que la inclusión de la tecnología es cada vez más frecuente en todos los deportes, particularmente en el fútbol.

A pesar de que Uruguay es un país chico, posee una vasta y exitosa historia, tanto a nivel continental como a nivel mundial, y sumado a que un gran porcentaje de la gente es aficionada a este deporte, cobra gran importancia tener un campeonato de fútbol que sea atractivo para el espectador. De todas formas, si bien es muy importante que el torneo sea atractivo, hay otros factores claves a la hora de organizar un campeonato, como que sea justo deportivamente para todos los equipos participantes, sin importar la magnitud que tenga cada uno de ellos, o que presente procesos transparentes en todas las decisiones tomadas, para así evitar la mayor cantidad de polémicas posibles. Estas consideraciones, junto con muchas otras posibles, pueden ser vistas como algo no cuantificable, o que no existe la posibilidad de ser logradas, total o parcialmente, por medio del uso de la tecnología, y es aquí en donde entra en juego la problemática que se buscará solucionar a lo largo de este proyecto: la elaboración de un fixture para el campeonato de fútbol uruguayo de primera división, utilizando técnicas y/o algoritmos de resolución de problemas de optimización. De esta forma se busca lograr que se alcance, o al menos en los papeles, un torneo más justo, equitativo, atractivo y transparente en todo su

proceso organizativo.

Un fixture puede definirse como el calendario de partidos a disputarse en un determinado campeonato de cualquier índole, siendo fútbol en este caso particular. Incluye además de quién disputa cada partido, el lugar, las fechas y horas de cada uno de ellos para cada jornada del campeonato.

Históricamente, en el fútbol uruguayo en general, los fixtures han sido elaborados mayoritariamente a través de un sorteo, presentando ciertas particularidades, por ejemplo, fijando ciertos partidos para una fecha específica, intentando evitar que dichos partidos se disputen en las primeras fechas del campeonato, derivando en consecuencias negativas, tanto deportiva como económicamente. Sin embargo, existe una serie de problemáticas que no pueden ser atacadas siguiendo esta estrategia, e incluso es altamente complejo realizarlo por sorteo o manualmente por personas dedicadas a ello. Por tanto, es de interés contar con un mecanismo de generación de los fixtures, pero que además pueda cumplir con las restricciones existentes, sumado a un conjunto amplio de nuevas restricciones y limitantes.

El uso de técnicas de optimización cada vez está cobrando un papel más importante en diversos ámbitos de la vida cotidiana, y la elaboración de fixtures para diversos campeonatos deportivos no es la excepción. Al planear un viaje o nuestro día, cada vez que una empresa toma una decisión importante de negocio o que un equipo de ingenieros civiles evalúa cómo manejar el tráfico de una ciudad, se está usando la optimización. En todas estas actividades, se tiene un objetivo que se quiere optimizar como ser: costo, ganancia, calidad, rendimiento, entre otros. Cualquiera sea el problema en la vida real, se podrían analizar sus características y con ellas armar un modelo de optimización con distintos objetivos, variables y restricciones para recomendar una solución que permita obtener el mejor resultado posible.

Un primer enfoque para resolver problemas de optimización es la Programación Lineal, en donde, valga la redundancia, se busca optimizar una función objetivo lineal, cumpliendo una serie de restricciones. Sin embargo, la mayoría de los problemas del mundo real son no-lineales, y muchas veces con restricciones que colisionan entre sí, por lo que encontrar una solución óptima se puede

tornar mucho más difícil, o incluso imposible. A medida que los problemas comienzan a volverse más complejos, es necesario buscar otros enfoques, más allá de los métodos exactos, y aquí es donde entran en juego los métodos metaheurísticos. Estos métodos son, básicamente, algoritmos heurísticos generales que permiten la resolución de problemas de optimización.

Hasta ahora, se ha hecho referencia a modelados basados en un solo objetivo. Sin embargo, en el mundo real muchas veces es necesario lidiar con problemas que requiere optimizar varios objetivos al mismo tiempo. En estos casos estamos ante problemas de optimización multi-objetivo. Este tipo de problemas, a diferencia de los problemas de un solo objetivo, no necesariamente tienen una solución óptima que minimice todas las funciones objetivo simultáneamente. Muchas veces estos objetivos serán incompatibles entre sí, por ejemplo, minimizar el tiempo de un viaje y su costo asociado en ocasiones pueden ser difícil de cumplir al mismo tiempo, por lo que la solución óptima de uno de ellos podría empeorar el valor obtenido para otro. En estos casos, se tendrá que elegir cierto balance en la optimización de cada objetivo o incluso, si esto no es posible, se debe establecer ciertas preferencias entre los objetivos a cumplir. Para este tipo de problemas, también se suelen usar algoritmos metaheurísticos ya que han demostrado tener una buena efectividad.

El objetivo principal de este proyecto radica en la elaboración de un fixture para el campeonato de fútbol uruguayo, utilizando métodos de optimización, que permitan mejorar los fixtures que se utilizan normalmente en nuestro país en diversos factores. Por ejemplo, se podría aumentar la justicia deportiva, el atractivo del torneo y también las ganancias económicas para todos los entes involucrados, como son los equipos que disputan los campeonatos o las empresas de televisión, publicidad, etc. Sin embargo, es necesario transformar estos objetivos *abstractos* en algo cuantificable, que pueda ser utilizado de base para un modelado matemático y posteriormente en un algoritmo computacional.

Para ello, se tuvieron en cuenta una serie de objetivos cuantificables que buscan ser de ayuda a los aspectos presentados anteriormente. En primer lugar, para favorecer la justicia del fixture desarrollado, se buscará evitar que todos los equipos disputen partidos seguidos tanto de local como de visitante, evitando generar lo que se suele denominar breaks. Siguiendo con ello, en Uru-

guay se puede hacer una distinción dentro de los equipos por su nivel reciente o histórico, teniendo los equipos top por un lado, y el resto por otro. Por lo tanto, se buscará disminuir el denominado efecto arrastre, que se genera cuando un equipo se ve beneficiado al enfrentar a otro equipo que viene de disputar uno o más partidos recientemente contra equipos top, por el esfuerzo físico y mental extra que posiblemente conlleve enfrentarse a ellos. Al existir la posibilidad de que existan equipos ubicados geográficamente a distintas distancias entre sí, cobra interés también el hecho de minimizar o equiparar las distancias viajadas por cada equipo, hecho que inexorablemente genera un desgaste físico importante. Así como éstos, pueden existir infinidad de posibles objetivos, que elaborando un fixture por sorteo o de manera manual, se tornan imposibles de considerar, por lo que la problemática que se buscará resolver en este proyecto puede ser a posteriori muy beneficiosa para la Asociación Uruguaya de Fútbol.

En primera instancia, y luego de haber relevado y transformado estos requerimientos, objetivos y restricciones en elementos cuantificables, es decir, en un modelo matemático, se consideraron dos enfoques. Por un lado, y tomando como base el modelo anterior, se abordó el problema utilizando un solver de Programación Entera, de forma de conseguir optimizar satisfactoriamente cualquiera de los objetivos por separado. Por otra parte, para poder tener en cuenta todos los objetivos simultáneamente, fue diseñado un algoritmo evolutivo multi-objetivo, particularmente un algoritmo genético NSGA-II, que pudo brindar un mayor abanico de posibilidades a la hora de desarrollar un fixture.

Cabe destacar que todas las pruebas realizadas con los algoritmos diseñados fueron comparadas directamente con fixtures reales del Campeonato Uruguayo de Fútbol, incluyendo también algunos fixtures correspondientes a la Liga Uruguaya de Básquetbol. El básquetbol fue incluido debido a sus semejanzas con el fútbol, en lo que refiere a su fixture, y porque se analizaron los fixtures utilizados en los últimos años y presentaban problemáticas similares a las del fútbol.

Los resultados obtenidos, tanto para el fútbol como para el básquetbol, demostraron que es posible obtener fixtures con resultados mejores a los utilizados al día de hoy en los campeonatos reales. Esto muestra que en Uruguay puede ser útil el uso de técnicas de optimización a la hora de resolver proble-

mas de esta índole.

El presente informe se encuentra estructurado de la siguiente manera. En primer lugar se presenta una introducción teórica con todos los conceptos que son necesarios para tener un conocimiento general de la realidad del fútbol uruguayo, en particular de cómo se desarrolla y estructura el campeonato local, siguiendo con una lista de conceptos clave que son necesarios para entender de manera adecuada el desarrollo de este proyecto. El mismo capítulo incluye también un relevamiento vasto de bibliografía con el objetivo de conocer el estado del arte a nivel mundial en cuanto a este tema, tanto para el fútbol, como para otros deportes. El siguiente capítulo consta del modelado matemático del problema, el cual fue desarrollado como un proceso incremental de tres fases, en donde en la segunda y en la tercera se tomó como entrada el resultado de la fase anterior, agregando nuevas especificaciones y consideraciones. Una vez presentado el modelado, se presenta un capítulo dedicado a los algoritmos multi-objetivo, definiendo una serie de conceptos importantes relativos a ellos y también detallando el algoritmo diseñado en este proyecto. Luego, se realizó un detallado análisis de los resultados obtenidos, tanto para el algoritmo exacto, como para el multi-objetivo, utilizando distintas métricas y comparando con los fixtures reales que han sido utilizados en los últimos años. Para finalizar, se presentan las conclusiones, posibles mejoras y trabajo futuro para el trabajo desarrollado.

Capítulo 2

Conceptos preliminares y estado del arte

Antes de adentrarse plenamente en cómo se implementó una solución que permita elaborar un fixture para el campeonato antes descrito, es importante tener una base de conocimiento introductoria, que permita ayudar a entender la realidad planteada, y la solución a la misma. Para ello, el comienzo de este capítulo consta de una presentación del campeonato uruguayo de fútbol al día de hoy, incluyendo el formato del mismo y la disposición de los equipos participantes. Por otra parte, existen una serie de conceptos sobre la elaboración de fixtures en general, que van a ser utilizados a lo largo del informe con frecuencia, y que es importante definirlos previamente. Por último, una etapa clave previa a comenzar el modelado de la posible solución, es el relevamiento del estado del arte para la problemática que se busca resolver. Se estudió y relevó una serie de artículos científicos, incluyendo algunos sobre otros deportes, con el objetivo de tener un conocimiento que abarque mayores posibilidades para el problema a resolver. En este capítulo se presentará un resumen de los que fueron considerados los artículos más valiosos y que fueron de mayor utilidad para esta investigación.

2.1. Conceptos preliminares

Para empezar a diagramar una posible solución para la realidad del fútbol uruguayo es de interés conocer el formato de este campeonato en profundidad.

Cada año, este torneo es disputado por dieciséis equipos. Al final del torneo tres de ellos descienden a la segunda categoría, dando lugar a que los tres mejores de ella asciendan para disputar el torneo de primera división al año siguiente. El campeonato se puede dividir a grandes rasgos en dos torneos cortos, denominados Apertura y Clausura. Cada uno de ellos se juega siguiendo una modalidad de todos contra todos, con una determinada localía en cada partido, es decir, si se considera la abstracción de un partido como A-B, siendo A y B dos equipos participantes de la competición, y dicho partido disputándose con el equipo A en condición de local en su estadio. Una vez jugados todos los partidos, el equipo que consiga más puntos será declarado el campeón de dicho torneo corto, existiendo la posibilidad de igualdad de puntos entre dos equipos, en donde se puede jugar un partido extra para definir el campeón. Para el caso del torneo Clausura, hasta el día de hoy, se mantiene el mismo formato que en el torneo Apertura, así como también el mismo orden de partidos para las quince fechas a disputarse, con la salvedad que se invierten las localías. Es decir, volviendo al ejemplo anterior, asumiendo en la fecha n se juega el partido A-B en el torneo Apertura, entonces se disputará el partido B-A en la misma fecha del torneo Clausura.

Con el objetivo de darle un mayor rodaje a los equipos, a partir del año 2017 se empezó a disputar el torneo Intermedio, un nuevo torneo corto, a disputarse entre el Apertura y Clausura de cada año. En el torneo Intermedio se separan los dieciséis equipos en dos grupos de ocho, definidos con el criterio de que los ocho equipos que quedaron en posiciones impares del torneo Apertura van al mismo grupo, y por consiguiente los que quedaron en posición par van al otro grupo. Definidos los grupos, se juega en un formato todos contra todos, de la misma manera que se explicó anteriormente, y luego los dos ganadores de cada grupo se enfrentarán en una final, que define el campeón del torneo intermedio. Como beneficio, el campeón de dicho torneo obtiene un pase a las copas internacionales a disputarse al año siguiente.

Como se explicó anteriormente, hay muchos conceptos que serán utilizados en gran parte de este informe y que es necesario conocer para entender este trabajo, desde lo más general, como por ejemplo en qué consiste un fixture y cuáles son sus principales componentes, hasta aspectos más detallados que involucren características específicas de un tipo de fixture en particular. A con-

tinuación, se presenta un listado con los conceptos más importantes:

- **División (deportiva)** - Agrupamiento de los equipos, dado por el rendimiento obtenido en el campeonato anterior. En el caso de Uruguay, existen 3 divisiones, donde actualmente la primera división cuenta con 16 equipos, la segunda con 12 y la tercera con 18.
- **Fecha o round** - Día o grupo de días en los que se juega una jornada de partidos en un torneo.
- **Break** - Secuencia de partidos consecutivos jugados en condición de local o en condición de visitante. Esto suele tener bastante importancia en países con gran superficie geográfica, ya que los equipos que jueguen partidos consecutivos de visitante, probablemente tengan que viajar grandes distancias entre partidos causando una fatiga adicional en los jugadores.
- **Pseudo-break** - Ocurre cuando un equipo juega dos fechas seguidas de visitante pero uno de estos dos partidos es en su misma provincia o región. Es un hecho poco común y un ejemplo del mismo puede ser, si Nacional (cuyo estadio está ubicado en Montevideo) tiene que jugar una fecha de visitante contra Plaza Colonia (cuyo estadio está ubicado en el departamento de Colonia) y la fecha siguiente también de visitante contra Peñarol (cuyo estadio está ubicado en el departamento de Montevideo). Este concepto no tiene tanta relevancia en el campeonato uruguayo, dado que la mayoría de los equipos pertenecen a un mismo departamento (Montevideo).
- **Efecto arrastre** - Fatiga en los jugadores causada como consecuencia de jugar de forma consecutiva contra equipos muy fuertes.
- **Road trips** - Conjunto de partidos consecutivos jugados en condición de visitante.
- **Home stands** - Conjunto de partidos consecutivos jugados en condición de local.
- **Pair clashes** - Encuentros jugados en simultáneo en estadios ubicados geográficamente muy cerca, los cuales pueden ocasionar incidentes.
- **Equipos top** - Equipos con un gran historial de campeonatos ganados, los cuales normalmente se posicionan en la cima de la tabla de puntos.
- **Partidos top** - Partidos disputados entre equipos top.
- **Round robin** - Torneo en el cual cada equipo juega contra cada uno de

los equipos una única vez.

- **Doble round robin** - Torneo en el cual cada equipo juega dos veces contra cada uno de los equipos.
- **Esquema espejo** - Torneo doble round robin el cual se divide en dos partes, en las cuales se juegan los mismos partidos, en el mismo orden pero con la localía invertida. Actualmente, el campeonato Uruguayo (obviando el torneo intermedio) sigue este esquema. En la siguiente Tabla se muestra un ejemplo de un fixture con esquema espejo de 4 equipos.

Fecha 1	**Fecha 2**	**Fecha 3**	**Fecha 4**	**Fecha 5**	**Fecha 6**
E1 - E2	E3 - E1	E1 - E4	E2 - E1	E1 - E3	E4 - E1
E3 - E4	E2 - E4	E3 - E2	E4 - E3	E4 - E2	E2 - E3

- **Esquema Francés** - Consiste en un esquema espejo donde la que sería la primera fecha de la segunda vuelta se mueve para el final – lo que da como resultado que el fixture tenga los mismos partidos pero con la localía invertida en la primera y en la última fecha. Este tipo de esquema es utilizado por ejemplo, en la Copa Libertadores para la elaboración del fixture de la fase de grupos, en donde la totalidad de los equipos se divide en grupos de cuatro cada uno, enfrentándose todos contra todos en dos oportunidades.

Fecha 1	**Fecha 2**	**Fecha 3**	**Fecha 4**	**Fecha 5**	**Fecha 6**
E1 - E2	E3 - E1	E1 - E4	E1 - E3	E4 - E1	E2 - E1
E3 - E4	E2 - E4	E3 - E2	E4 - E2	E2 - E3	E4 - E3

  En el ejemplo se puede notar como la Fecha 1 y 6 son iguales pero con localía invertida, así como la Fecha 2 con la 4, y la Fecha 3 con la 5.
- **Partido clásico** - Partido disputado entre dos equipos pertenecientes a una misma ciudad o región entre los cuales existe una gran rivalidad.

2.2. Relevamiento del estado del arte

Una vez definidos los conceptos anteriores, ya se está en condiciones de seguir con la siguiente etapa, la alusiva al estado del arte actual. Es importante conocer en qué estado y qué posibilidades, se encuentran para resolver el

problema. Para ello, se relevó una serie de artículos científicos sobre soluciones creadas para la elaboración de fixtures a través de técnicas de optimización para distintos campeonatos y con distintas necesidades. Se tuvieron en cuenta soluciones aplicadas en países latinoamericanos, así como también en otras partes del mundo como Australia y Bélgica. Es importante agregar que además se incluyeron artículos de otros deportes como vóleibol y básquetbol, los que se consideraron interesantes a la hora del relevamiento para tener una visión más amplia abarcando otros deportes, aún cuando el formato de estos torneos puede ser similar al caso del fútbol. Esto es de suma utilidad ya que provee un panorama general de lo que se puede lograr y también de algunas limitaciones que eventualmente se pueden encontrar, utilizando distintas metodologías, que pueden ser perfectamente aplicables para el caso del fútbol uruguayo.

Es importante destacar también que en base a la información recabada en este relevamiento sobre el estado del arte y, particularmente, en los artículos considerados, se puede observar que en la amplia mayoría de los casos uno de los objetivos a cumplir es que cada equipo recorra la menor distancia posible a lo largo del campeonato. Este tipo de problema suele resolverse utilizando como base el Traveling Tournament Problem (o TTP por sus siglas en inglés). Este problema busca, principalmente, minimizar las distancias recorridas por cada equipo. Más específicamente, el objetivo del TTP es, dado un conjunto de equipos, elaborar un torneo doble round robin en donde el máximo número de partidos seguidos de local o de visitante para cada equipo esté acotado y además la distancia recorrida por los equipos sea minimizada. Formalmente, el TTP se define como (Thielen and Westphal, 2011):

Sean:

- T un conjunto de equipos donde $|T| = n \geq 4$ y n es par
- Una matriz de distancia $D_{N \times N}$ simétrica, donde cada elemento $d_{ij} \geq 0$ representa la distancia entre las localidades de los equipos i y j. Además se cumple la desigualdad triangular y que $d_{ii} = 0 \, \forall \quad i$
- Un entero no negativo l que determina el largo máximo de cada break

Entonces, se busca encontrar un torneo doble round robin de los equipos en T tal que:

- La distancia total recorrida por los equipos sea minimizada

- Para cada equipo, el largo máximo de un break de local es menor a l
- Para cada equipo, el largo máximo de un break de visitante es menor a l

Resolver un problema de este estilo es computacionalmente difícil, debido a que se tiene que calcular el grado de factibilidad de los distintos patrones de local-visitante para poder evitar largos *home stands* y *road trips* y además se debe optimizar la distancia total recorrida por cada equipo. Estos dos problemas pueden ser resueltos de manera fácil individualmente a través de programación con restricciones o programación entera. Sin embargo, la combinación de ambos puede llegar a ser un problema muy complejo de resolver usando únicamente uno de ellos, por lo que se suelen usar combinaciones de ambos (Easton et al., 2001). Más específicamente, se ha demostrado que resolver un problema de este estilo es NP-completo (Thielen and Westphal, 2011).

A continuación se presentan los artículos mencionados anteriormente, con un breve resumen de cada uno de ellos, agrupados en dos categorías. En primer lugar, todos los artículos sobre fútbol y luego los alusivos a otros deportes. Para ambos casos se optó por presentarlos siguiendo el ordenamiento cronológico basado en la fecha de publicación de cada artículo. Por último, la estructura seguida para la elaboración de cada resumen es la siguiente: una breve introducción del formato del campeonato, cantidad de equipos, las restricciones y el/los objetivo/s que se debieron contemplar a la hora de generar el fixture, el o los métodos elegidos para la solución, y por último los resultados obtenidos, tanto cuantitativos como cualitativos.

2.2.1. Relevamiento de artículos sobre fútbol

2.2.1.1. Scheduling the Belgian Soccer League (Goossens and Spieksma, 2009)

El campeonato de fútbol de Bélgica está compuesto por 18 equipos, siguiendo en general un esquema de doble round robin, jugando todos contra todos en dos oportunidades, siendo casi la totalidad de los partidos disputados los fines de semana.

La restricción más importante a considerar en este torneo es que cada equipo puede jugar hasta un máximo de dos partidos consecutivos, ya sea de local

o de visitante, es decir, un break. A su vez, se intenta minimizar la cantidad total de dichos acontecimientos. Por otra parte, algunas restricciones importantes manifestadas por la confederación, autoridades locales y compañías de televisión que fueron consideradas son: evitar que equipos de la misma localidad/ciudad jueguen de local en el mismo día, ya sea por un tema de seguridad o para no disminuir la cantidad de espectadores de cada partido. También, para darle un mayor atractivo al torneo, se fuerza a que cada equipo reciba por lo menos a uno de los equipos considerados top en cada mitad del torneo y evitar a su vez que un equipo menor juegue varios partidos seguidos contra equipos importantes, por el cansancio físico y mental extra que ello supone. Por último, también se han considerado las semanas que existen partidos por competiciones europeas entre semana, creando un fixture que permita a los equipos participantes de ellas llegar con mayor descanso a sus respectivos partidos.

Para la resolución del problema en este caso se siguió un enfoque de programación lineal entera, en donde se agruparon las restricciones en cinco niveles de prioridad, cada uno de ellos con un nivel de penalización por cada restricción que no se cumpliese, con el objetivo de que el mejor resultado obtenido sea el que resulte en una sumatoria de penalizaciones menor.

Los resultados obtenidos fueron considerados ampliamente satisfactorios, obteniéndose un valor alrededor de siete veces menor en la sumatoria de las penalizaciones, si se compara la elaboración del nuevo fixture de manera automática con los anteriores que eran creados manualmente por la organización. Con respecto a la cantidad de restricciones cumplidas se observó una diferencia similar. Todo esto significó la creación de un fixture más atractivo y justo para todos los equipos, derivando en que en años posteriores se fueran incorporando nuevas restricciones, e incluso se empezó a adoptar este formato para los torneos de las divisiones menores en dicho país.

2.2.1.2. Scheduling the professional Ecuadorian football league by Integer Programming (Recalde et al., 2013)

Al igual que en nuestro país, el fútbol es el deporte más popular en Ecuador. El formato del torneo consta de dos divisiones (A y B), compuesta cada una por doce equipos, los cuales compiten bajo las mismas condiciones. Cada división es disputada en tres fases, donde las primeras dos, constan en un doble round robin, intercambiando las localías e invirtiendo el orden de los encuentros en la segunda ronda de cada fase, es decir, siguiendo un esquema espejo invertido. Por último, la tercera y última ronda, enfrenta a los equipos que obtuvieron el primer lugar en las primeras dos fases para definir el equipo campeón.

Debido a que en Ecuador existen provincias con más de un equipo, una condición deseable para la federación es que al momento de que un equipo juegue dos partidos seguidos de visitante, uno de ellos sea en su misma provincia, considerado esto como una medida de equilibrio. Otra condición deseable es evitar que los partidos considerados clásicos se disputen tanto en las primeras dos o en las últimas dos fechas del torneo, dado que la disputa de dichos partidos cerca de la mitad del torneo, los hace más atractivos y pueden generar más ingresos económicos. Por último, también se busca tanto minimizar la cantidad de breaks, cómo maximizar la cantidad de pseudo-breaks en el torneo.

Se utilizaron dos enfoques distintos para la resolución del problema en cuestión, uno en una única fase, utilizando un solo conjunto de restricciones formado por la totalidad de las restricciones anteriormente mencionadas y otro utilizando una heurística de tres fases; la primera de ellas para generar patrones factibles, la segunda para asignarlos a grupos determinados y la última para generar un calendario global factible. Se pudo notar una gran diferencia de performance entre ambos enfoques incluso logrando maximizar la cantidad de pseudo-breaks de igual forma (ocho en ambos casos), ya que el tiempo de resolución del segundo enfoque (menos de un segundo) fue ampliamente más rápido que el tiempo de resolución del primero (un par de minutos).

A modo de análisis de resultados, se pasó de un fixture con dieciséis breaks y cuatro pseudo-breaks utilizando el enfoque tradicional, a uno con 12 breaks

y 8 pseudo-breaks. Por lo que se puede concluir que la utilización de este tipo de métodos de optimización tuvieron un resultado altamente positivo, ya que el fixture propuesto fue aprobado por la Federación Ecuatoriana de Fútbol.

2.2.1.3. Scheduling the English Football League with a Multi-objective Evolutionary Algorithm (While and Kendall, 2014)

La liga inglesa de fútbol es una liga que cuenta con cuatro divisiones, donde los equipos suben y bajan de división dependiendo del posicionamiento de los mismos al terminar los campeonatos. Estas cuatro divisiones están compuestas por English Premier League (EPL), Championship, League 1, y League 2.

Por un lado, la EPL cuenta con 20 equipos, mientras que las otras tres divisiones tienen 24 cada una. Los cuatro campeonatos son doble round robin, por lo que 2036 partidos requieren ser programados en cada temporada, lo que representa un gran reto para los organizadores. Esto podría ser encarado de forma individual por división, ya que ningún equipo juega contra otros equipos de otra división. Sin embargo, existe una dependencia entre los equipos de distintas divisiones ya que muchos de ellos comparten localía en la misma ciudad, por lo cual podrían darse diversos tipos de incidentes si muchos equipos de una misma región juegan al mismo tiempo como locales.

Por lo tanto, a la hora de organizar un fixture para esta competición, se buscó minimizar la cantidad de partidos disputados simultáneamente en estadios muy cercanos (pair clashes). Por ejemplo, para el caso de Londres, se buscó que sólo seis de los equipos que ahí ejercen la localía puedan jugar de local en una misma fecha. Además, fue también considerado el objetivo de minimizar la distancia recorrida por cada equipo.

Para resolver la creación del fixture se decidió usar un algoritmo evolutivo multi-objetivo. Esto produjo varios beneficios, los cuales se pueden notar al analizar los resultados obtenidos. Por ejemplo, en la temporada 2002-2003 se puede ver que una de las soluciones obtenidas permite reducir el costo de viaje de los equipos en un 3 % aunque existan 15 pair clashes, ya que los organiza-

dores consideraron que esa cantidad de pair clashes no es de gran riesgo y es más conveniente aprovechar ese porcentaje de ahorro en los costos de viajes.

2.2.1.4. Scheduling the South American Qualifiers to the 2018 FIFA World Cup by Integer Programming (Durán et al., 2017)

Las eliminatorias de CONMEBOL para el mundial de fútbol son disputadas cada cuatro años y cuentan con 10 equipos sudamericanos compitiendo por cinco lugares en la copa mundial FIFA, de los cuales cuatro clasifican en forma directa y el quinto deberá jugar un repechaje contra otro país de otra confederación para poder lograr alcanzar el pase al campeonato mundial.

La duración de las eliminatorias es de dos años y en cada doble fecha del torneo se juegan dos partidos por equipo separados por pocos días entre sí, es decir, cada doble fecha consta de 10 partidos. Al ser tan larga la duración del torneo y al jugarse doble fechas, la separación entre dichas doble fechas en ocasiones es de varios meses.

Para construir este fixture, se decidió utilizar un esquema francés, que consiste en un esquema espejo donde la que sería la primera fecha de la segunda vuelta se mueve para el final – lo que da como resultado que el fixture tenga los mismos partidos pero con la localía invertida en la primera y en la última fecha.

Con respecto a las restricciones, el mayor foco a la hora de construir este fixture fue el minimizar la cantidad de breaks fuera de casa. En el contexto de esta competencia en la cual cada fecha consta de una doble jornada, un break se refiere a que un equipo juega los dos partidos de la fecha de local o visitante. Por lo tanto, la presencia de un break en los cuales ambos partidos se juegan de visitante implica que ciertos equipos estén varios meses sin jugar en su país, lo cual podría impactar negativamente el nivel anímico y económico debido a la menor recaudación en la venta de entradas de éstos y por ende perjudicando sus chances de clasificar a la Copa Mundial de Fútbol.

Por otro lado, también se buscó balancear para cada equipo las fechas en

las que éstos juegan el primer partido de local y el segundo de visitante. Esto es importante ya que hay una tendencia de parte de los equipos a preferir este tipo de fechas – en lugar de jugar primero de visitante y luego de local. Esto es, por ejemplo, porque los equipos tienen más tiempo para prepararse para el partido, ya que todos los jugadores viajan a su país de origen en principio independientemente de si la fecha se jugará de local o visitante; así como también da la posibilidad a los jugadores de volver a sus clubes después de jugar el segundo fuera de su país, reduciendo los tiempos de viaje. En fixtures previos se pudo notar un desbalance en este aspecto (equipos con muchas fechas donde jugaban el primer partido de visitante y el segundo de local y pocas fechas del caso contrario, y viceversa), por lo que los autores consideraron este balance como un buen objetivo para intentar lograr un fixture más justo.

Otra restricción que los autores decidieron añadir fue la de prevenir que los equipos jueguen partidos consecutivos contra equipos fuertes, aunque no especifican el criterio para considerar cuál equipo es fuerte y cuál no.

Para resolver este problema, los autores optaron por un modelo de programación entera. El fixture propuesto fue aprobado de forma unánime por los miembros de la CONMEBOL y fue utilizado por primera vez en las eliminatorias del mundial de Rusia 2018. Además, el objetivo de minimizar la cantidad de breaks resultó en un fixture para el cual todos los equipos jugaban al menos una vez en casa en cada doble fecha, a diferencia de años anteriores, dónde se tenían 18 breaks por fixture.

2.2.1.5. Scheduling the Australian Football League (Kyngäs et al., 2017)

El fútbol en Australia se ha convertido en el deporte más popular en los últimos años. El torneo disputado en dicho país consta de un round robin simple, sumado a cinco partidos extras, que tienen la particularidad de desarrollarse durante el transcurso del propio round robin, con el objetivo de que exista una mayor cantidad de partidos disputados, junto con darle al torneo un mayor atractivo desde el punto de vista comercial.

A la hora de establecer las restricciones, los autores tuvieron en cuenta que hay varios equipos que no poseen un estadio fijo para jugar sus partidos de local y también que existen estadios que alojan partidos de local de varios equipos, por lo que habrá que decidir en qué estadio se jugará cada partido. Por otra parte, dado que Australia es un país grande, la minimización de las distancias viajadas por cada equipo es otro aspecto principal a ser tenido en cuenta en las restricciones, así como también la minimización de cantidad de breaks, lo cual ayuda a obtener un torneo más justo.

Para la resolución de este problema los autores tomaron un enfoque de programación basado en restricciones, cuya resolución se hace en tres fases utilizando la heurística PEAST. Para la primera etapa se deciden las localías y oponentes para cada equipo, en la segunda se asignan dichos partidos a fechas específicas, y por último a cada partido se le asigna un estadio, día y hora para ser disputado.

A modo de comparación con el fixture elaborado el año anterior, se obtuvieron importantes ventajas. En primer lugar, utilizando este enfoque basado en restricciones, se pudo obtener un fixture que cumpla con todas ellas, lo que no sucedió el año anterior, en donde cuatro restricciones no se cumplían. Respecto a los dos objetivos primordiales, se logró disminuir significativamente las distancias viajadas por los equipos, así como también se redujo la cantidad de breaks de 94 a 74 en la solución obtenida en este trabajo.

2.2.1.6. Operations Research Transforms the Scheduling of Chilean Soccer Leagues and South American World Cup Qualifiers (Alarcón et al., 2017)

La Asociación Nacional de Fútbol Profesional (ANFP - Chile) desde 2005 ha utilizado técnicas de optimización para la elaboración de los fixtures de todos los torneos desarrollados, generando grandes beneficios tanto económicos (superiores a 59 millones de dólares) como no económicos.

Existe una serie de requerimientos, los cuales fueron tomados en cuenta para encontrar una solución factible a este problema. Dado que Chile es un

país geográficamente angosto y largo, es de interés minimizar las distancias viajadas por los equipos, para generar un ahorro económico y una disminución en la fatiga de los jugadores. El clima también juega un papel importante, por lo que se busca evitar jugar muchos partidos en localidades lluviosas, evitando así la suspensión de partidos, así como también se busca disputar los partidos más atractivos en las zonas más turísticas del país. La televisión siempre tiene un gran peso, y dado que sus periodistas tienen que viajar de estadio a estadio, es de interés ubicar partidos atractivos lo más cercano posible en cada fecha. En lo que respecta a seguridad se prefiere evitar que equipos con estadios cercanos jueguen de local la misma fecha para evitar disturbios. Por último, es deseable que el torneo sea lo más atractivo posible, para esto se le asigna un puntaje a cada partido dependiendo de la ronda en que se juegue, y se busca disputar los de mayor puntaje en la segunda mitad del torneo.

El problema en cuestión fue modelado como un problema de programación lineal entera, donde la estrategia de resolución se basó en encontrar en primer lugar una solución factible a través de un enfoque basado en restricciones, resolviendo el TTP, y luego mejorando con respecto a la función objetivo a partir de esta primera solución. Para esta mejora, se eligió un enfoque de descomposición basada en patrones. Se utilizó un enfoque heurístico en tres etapas, en primer lugar, utilizando un modelo de programación lineal entera se construye un conjunto de N patrones (N equipos), luego se toma un grupo de estos patrones y se asigna a determinados equipos siguiendo los requerimientos apropiados, dejando el resto sin asignar. Por último, el modelo final asigna al resto de los equipos un patrón que cumpla los requerimientos del problema.

La solución obtenida logró generar beneficios tanto cuantitativos como cualitativos. Entre ellos, su puesta en práctica generó un incremento de un 74 % en la asistencia a los estadios, disminuyeron los costos de transmisión de los partidos, así como se obtuvo un ahorro en los traslados de los equipos. También se pudo generar un torneo más atractivo, con jugadores menos fatigados y con mayor transparencia a la hora de su planificación.

2.2.2. Relevamiento de artículos sobre otros deportes

2.2.2.1. An application of the traveling tournament problem: The Argentine volleyball league (Bonomo et al., 2012)

La primera división argentina de vóleibol es un torneo que consta de 12 equipos, y tiene dos fases. La primera de ellas, llamada la fase regular, consiste en un doble round robin, en la cual los mejores ocho equipos clasifican a los playoffs. La segunda fase es de eliminación directa con fechas de enfrentamiento donde clasifica el mejor en un máximo de cinco partidos, a excepción de la final que es al mejor de siete. Los 12 equipos participantes se dividen en seis parejas, cada una de ellas debido a cercanías geográficas. En esta liga, los partidos se juegan los jueves y los sábados, donde los partidos jugados en estos dos días de una misma semana forman un weekend. Cada weekend, una mitad de las parejas visita a la otra mitad, donde los equipos de la pareja visitante juegan contra los equipos de la pareja local. Sin embargo, existen dos weekends especiales, llamados intra-weekends en donde los equipos de cada pareja se enfrentan entre ellos. Por tanto, cada equipo disputará 22 partidos en un lapso de 12 weekends.

A modo de restricciones, se pueden considerar varias con distintos grados de importancia, como por ejemplo, que ciertos equipos no estén en la misma pareja para evitar grandes viajes de jueves a sábado, mantener equipos top en parejas distintas para darle un mayor atractivo al torneo, minimizar la distancia de viaje total y la diferencia entre los equipos que más y menos viajan a lo largo del torneo, evitar que una pareja juegue muchos partidos seguidos de visitante, entre otras.

La resolución del fixture para esta competición consta de dos problemáticas principales, por un lado cómo definir las parejas de equipos, y por otro, cómo definir los enfrentamientos entre cada una de las parejas. La manera de tomar esa decisiones fue aplicando un modelo de programación lineal entera que resuelva ambos problemas al mismo tiempo. Este método resulta ser eficiente, en el sentido de que ambas etapas pueden ser resueltas de manera muy cercana a la optimalidad, y siendo a su vez fácil de traspasar a las autoridades del torneo.

Los resultados obtenidos fueron valorados como muy positivos. En cuestiones cuantitativas, se redujo la distancia total recorrida en un 20 %, existiendo un ahorro de US$60.000 en gastos por viaje anuales, también derivando en un mayor tiempo de descanso para los jugadores participantes. También se redujo la diferencia entre los equipos que más viajaron en contraste con los que menos lo hicieron. Por último, los dirigentes del torneo demostraron un alto grado de conformismo por la facilidad que el sistema permite para agregar nuevas restricciones.

2.2.2.2. Scheduling Argentina's professional basketball leagues: A variation on the Travelling Tournament Problem (Durán et al., 2019)

El básquetbol es uno de los deportes más populares en Argentina, luego del fútbol, por lo que ha surgido la necesidad de poder obtener mejores fixtures, que permitan por ejemplo, reducir los tiempos de viaje de los equipos al jugar de visitante. Para la temporada 2015-2016, la primera división estuvo compuesta por 20 equipos y la segunda por 26. Para ambos casos el torneo se dividió en dos fases, una regional, en donde se dividen los equipos en dos grupos (siguiendo criterios geográficos de cercanía) y luego una fase nacional con la totalidad de los equipos.

Dado que la restricción más importante en este problema es la minimización de los viajes y el costo asociado a los mismos, la Asociación Argentina de Básquetbol solicitó a cada equipo que brinde una cantidad redundante de viajes preferidos, para poder incorporarlos al modelo y así poder obtener una solución factible de manera más fácil. También se consideraron otras restricciones, para evitar seguidillas extensas de partidos de local o visitante, evitar partidos de local cuando el estadio no está disponible para ser utilizado, requerimientos específicos de los canales de televisión, y también programar partidos atractivos en etapas claves dentro de cada torneo.

Para obtener la solución de este problema se utilizó como base el TTP, pero debido a que se tienen en cuenta las preferencias de los equipos y que además en el desarrollo del torneo puede haber fechas en donde algunos equipos no disputen ningún partido, el problema a resolver entra en la categoría TP-TR

TTP (Time Preferences-Time Relaxed Travelling Tournament Problem). El proceso de resolución se dividió en dos fases. En la primera etapa, se definió el conjunto de partidos para cada equipo. Debido a que puede ser un problema muy complejo computacionalmente, ya que el número de equipos puede ser incluso superior a veinte, se optó por dividir esta tarea en asignar los partidos por bloques. Por otro lado, en la segunda etapa se asignaron los partidos a los días específicos.

La solución obtenida proporcionó muy buenos resultados, entre ellos un gran descenso en la distancia viajada por los equipos, específicamente entre 20 % y 30 % menos de kilómetros promedio por viaje de visitante de cada equipo de la primera división. Esto derivó en un gran ahorro por parte de los equipos y en un importante descenso en la fatiga de los jugadores.

2.2.3. Resumen de los artículos relevados

A modo de síntesis de los artículos recién presentados, en la siguiente tabla se presenta un resumen con los aspectos más importantes de cada uno de ellos. Se detalla para cada artículo su año de publicación, el tamaño máximo de las instancias consideradas, el tipo de torneo y el enfoque elegido para la resolución del problema.

Artículo	**Año**	**Tamaño de instancias**	**Tipo de torneo**	**Enfoque de resolución**
Scheduling the Belgian Soccer League (Goossens and Spieksma, 2009)	2009	18 equipos	Doble round robin	Un algoritmo de dos fases en la que la primera establece los patrones local-visitante y la segunda donde se usa un algoritmo Tabú Search para minimizar la penalización por cada restricción no cumplida.
Scheduling the professional Ecuadorian football league by integer programming (Recalde et al., 2013)	2013	24 equipos	Dos series de 12 equipos donde cada serie juega tres fases: las primeras dos son un doble round robin con un esquema espejo invertido y la tercera consiste en dos partidos entre el mejor equipo de la primera fase y el mejor de la segunda para determinar el campeón.	Algoritmo de tres fases. En la primera fase, se genera un conjunto de patrones local-visitante factible. En la segunda fase, un subconjunto del conjunto factible generado en la fase 1 es asignado a cada grupo. Finalmente, en la última fase se crea un fixture doble round robin con esquema espejo invertido. Además también se utiliza programación lineal entera.

Scheduling the English Football League with a Multi-objective Evolutionary Algorithm (While and Kendall, 2014)	2014	92 equipos	Doble round robin	Algoritmo evolutivo multi-objetivo con siete objetivos. Cinco de los objetivos son en realidad restricciones, donde un valor demasiado alto indica una solución que viola la restricción.
Scheduling the South American Qualifiers to the 2018 FIFA World Cup by integer programming (Durán et al., 2017)	2016	10 equipos	Doble round robin	Programación lineal entera.
Scheduling The Australian Football (Kyngäs et al., 2017)	2016	18 equipos	Round robin con un adicional de 5 partidos extra por equipo	Algoritmo de tres fases. En la primera fase, se determinan los oponentes y los partidos de local de los equipos. La segunda fase asigna los partidos a las distintas fechas. Y por último, la tercera fase decide los horarios y lugares de inicio.

Operations research techniques for scheduling Chile's second division soccer league (Alarcón et al., 2017)	2017	14-16 equipos	Doble round robin	Programación lineal entera.
An application of the traveling tournament problem to the Argentine volleyball league (Bonomo et al., 2012)	2011	12 equipos	Fase regular estilo doble round robin donde los mejores ocho equipos clasifican a playoffs	Algoritmo de dos fases, donde la primera establece las parejas de equipos usando programación lineal entera y la segunda modela un TTP usando esas parejas.
Scheduling Argentina's professional basketball leagues - A variation on the Travelling Tournament Problem (Durán et al., 2019)	2018	18-26 equipos	Doble round robin	Programación lineal entera.

A partir de este relevamiento se pudo constatar que existen trabajos tanto regionales como internacionales que abordan la elaboración de un fixture para ligas con cantidades de equipos y características relativamente similares al Campeonato Uruguayo de Fútbol. Los resultados obtenidos en estos trabajos demuestran que es posible obtener fixtures mejores a los que suelen construirse sin el apoyo de técnicas de optimización.

Capítulo 3

Modelado del problema

En este capítulo, se detalla el modelado matemático que servirá de base para la implementación de un algoritmo que genere un fixture para el campeonato de fútbol uruguayo. El mismo está dividido en tres iteraciones incrementales, en donde cada iteración toma la anterior como entrada para considerar más aspectos que permitan obtener un fixture lo más completo posible.

En primer lugar, se presenta una primera versión, que permite modelar el torneo Apertura como un torneo round robin, considerando un conjunto acotado de restricciones. En la segunda iteración, se busca contemplar la totalidad del campeonato, es decir, tanto el torneo Apertura como el Clausura, agregando las restricciones necesarias para obtener un modelado matemático de las fechas. Por último, en la iteración final, el interés recae en considerar, además de las dos fechas, los días de cada una de ellas, generando un fixture que no sólo asigne los partidos a las fechas, si no que también cada partido se dispute en un día en particular.

3.1. Primera versión del modelo

Para esta primera iteración del modelado se incluyen las restricciones básicas que debe cumplir un fixture que busque seguir el formato de round robin en una única vuelta. Algunas de estas restricciones son, por ejemplo, que ningún equipo juegue contra sí mismo, que cada equipo juegue una vez contra cada equipo participante, y que además cada equipo juegue una única vez en cada fecha, no existiendo fechas libres para el caso del campeonato uruguayo. Lue-

go, se agrega el primer conjunto acotado de restricciones, compuesto por cinco restricciones específicas que se listan a continuación, junto con su formulación matemática, y que sirven como referencia para obtener los primeros fixtures.

Sea $E = \{1, ..., N\}$ el conjunto de equipos (N = 16), y E_T el subconjunto de equipos top ($E_T \subset E$). Además, sea $F = \{1, ..., 15\}$, el conjunto de fechas. Para $i, j \in E$ y $k \in F$, se define la variable binaria $x_{i,j,k}$ igual a 1 si el equipo i juega de local contra el equipo j en la fecha k, y 0 en otro caso.

3.1.1. Restricciones generales

- Ningún equipo puede jugar contra sí mismo.

$$x_{i,i,k} = 0 \quad \forall i \in E, \forall k \in F \tag{3.1}$$

3.1.2. Restricciones round robin

- Cada equipo deberá jugar, ya sea de local o visitante, una vez contra cada uno del resto de los equipos.

$$\sum_{k \in F} (x_{i,j,k} + x_{j,i,k}) = 1 \quad \forall i, j \in E : i \neq j \tag{3.2}$$

- Cada equipo deberá jugar un partido por fecha ya sea de local o de visitante.

$$\sum_{j \in E : j \neq i} (x_{i,j,k} + x_{j,i,k}) = 1 \quad \forall i \in E, \forall k \in F \tag{3.3}$$

3.1.3. Restricciones específicas

- Peñarol y Nacional no pueden enfrentarse en las primeras N fechas.

$$\sum_{k \in F : k \leq N} (x_{PEÑ,NAC,k} + x_{NAC,PEÑ,k}) = 0 \tag{3.4}$$

- Debe haber al menos P partidos entre equipos top en las últimas N fechas.

$$\sum_{i \in E_T} \sum_{j \in E_T : j \neq i} \sum_{k > |F| - N} x_{i,j,k} \geq P \tag{3.5}$$

- Equipos no top no pueden jugar más de N fechas seguidas contra equipos top.

$$\sum_{j \in E_T} \sum_{z=0}^{N} (x_{i,j,k+z} + x_{j,i,k+z}) \leq N \quad \forall k \in F : k \leq |F| - N, \forall i \in E \backslash E_T \tag{3.6}$$

- Esta restricción busca balancear la cantidad de partidos a jugar contra equipos top en cada mitad del campeonato para todos los equipos.

$$\sum_{k \in F : k \leq \lfloor \frac{|F|}{2} \rfloor} \sum_{j \in E_T : j \neq i} (x_{i,j,k} + x_{j,i,k}) \geq \frac{|E_T|}{5} + 1 \quad \forall i \in E \tag{3.7}$$

$$\sum_{k \in F : k > \lfloor \frac{|F|}{2} \rfloor} \sum_{j \in E_T : j \neq i} (x_{i,j,k} + x_{j,i,k}) \geq \frac{|E_T|}{5} + 1 \quad \forall i \in E \tag{3.8}$$

  La idea principal en este caso, es evitar que cada equipo juegue la mayoría de los partidos contra equipos top en la misma mitad del campeonato para así poder obtener un fixture más balanceado, y por lo tanto, más justo.

  La idea de dividir la cantidad de equipos top entre cinco y sumar uno surge de probar con distintas cantidad de equipos top y buscar un objetivo sea razonable. De esta forma, si se tuviese menos de cinco equipos top, se estaría restringiendo a tener al menos un partido en cada mitad. Del mismo modo, teniendo al menos cinco equipos top, esta restricción aseguraría al menos dos partidos contra cada equipo top en cada mitad. Se entiende que no es necesario considerar cantidades mucho más grandes de equipos top ya que no sería muy razonable (se asume un máximo de 25 % de equipos top).
- No puede haber más de un partido entre equipos top en la misma fecha.

$$\sum_{i \in E_T} \sum_{j \in E_T : j \neq i} x_{i,j,k} \leq 1 \quad \forall k \in F \tag{3.9}$$

3.1.4. Posibles funciones objetivo

- **Minimizar cantidad de breaks**

  Es conveniente notar que en los campeonatos con una cantidad par de

equipos, la cantidad de breaks de local es exactamente igual a la cantidad de breaks de visitante como se muestra en (Régin, 2001), por lo que alcanza sólo con minimizar una de ellas para minimizar el total.

Dicho esto, se definen las variables auxiliares $v_{i,k}$, las cuales tomarán el valor 1 en caso de que el equipo i tenga un break de local que comienza en la fecha k y 0 en caso contrario. Dichas variables deberán cumplir las siguientes restricciones:

$$\sum_{j \in E: j \neq i} (x_{i,j,k} + x_{i,j,k+1}) \leq 1 + v_{i,k} \quad \forall k \in F \backslash \{15\},\ \forall i \in E \tag{3.10}$$

$$\sum_{j \in E: j \neq i} x_{i,j,k} \geq v_{i,k} \quad \forall k \in F \backslash \{15\},\ \forall i \in E \tag{3.11}$$

$$\sum_{j \in E: j \neq i} x_{i,j,k+1} \geq v_{i,k} \quad \forall k \in F \backslash \{15\},\ \forall i \in E \tag{3.12}$$

La restricción de la ecuación 3.10 asegura que las variables deberán necesariamente tomar el valor uno si el equipo i tiene un break en la fecha k. Por otro lado, las restricciones de las ecuaciones 3.11 y 3.12 fuerzan a las variables a tomar el valor 0 si el equipo i no juega de local en la fecha k o en la siguiente $(k+1)$, respectivamente.

Por lo tanto, minimizar la cantidad total de breaks consiste en minimizar la siguiente función objetivo:

$$\text{mín}\, z = \sum_{i \in E} \sum_{k \in F \backslash \{15\}} v_{i,k} \tag{3.13}$$

- **Balancear el efecto arrastre entre todos los equipos**
  La idea para este objetivo es la de balancear entre todos los equipos, la cantidad de partidos en el campeonato en los cuales son beneficiados por el efectos de arrastre.

Por lo tanto, se necesita calcular para cada equipo la cantidad de partidos en los que su rival viene de jugar contra un equipo top. Para esto, se define la variable $a_{i,k}$ que tomará el valor 1 si el equipo i se beneficia del efecto arrastre en la fecha k y 0 en caso contrario, de la siguiente forma:

$$\sum_{j \in E: j \neq i} (\sum_{t \in E_T} (x_{j,t,k-1} + x_{t,j,k-1}) * (x_{i,j,k} + x_{j,i,k})) = a_{i,k} \quad \forall k \in F \backslash \{1\}, \forall i \in E \tag{3.14}$$

El objetivo de la ecuación anterior, radica en encontrar para cada par equipo-fecha $(i-k)$, otro equipo j, rival de i en la fecha k, que a su vez en la fecha anterior a k se haya enfrentado a un equipo top. Sólo en este caso ambos factores de la multiplicación serán iguales a uno, y por lo tanto se cumple que $a_{i,k} = 1$.

Luego, se define la variable a_i a la cual se le asigna la cantidad total de partidos a lo largo del campeonato en los cuales un equipo se benefició del efecto arrastre:

$$\sum_{k \in F: k \neq 1} a_{i,k} = a_i \quad \forall i \in E \tag{3.15}$$

Una vez sabido esto, el objetivo consiste en intentar minimizar el mayor valor del conjunto de a_i. Para conseguirlo, primero se necesita calcular el valor máximo del conjunto. Se define la variable a_{max} que está restringida por:

$$a_i \leq a_{max} \quad \forall i \in E \tag{3.16}$$

$$a_i + |F|b \geq a_{max} \quad \forall i \in E,\ b \in \{0, 1\} \tag{3.17}$$

donde b es una variable de decisión booleana.

Por lo tanto, la función objetivo queda de la siguiente manera:

$$\text{mín}\, z = a_{max} \tag{3.18}$$

- **Minimizar distancias viajadas**

  Otro objetivo que resulta interesante, sobre todo para los equipos de ciudades del interior, es el de minimizar las distancias viajadas por cada uno de los equipos. Este objetivo no sólo ayuda a que haya una mayor justicia a la hora de jugar los partidos, si no también permite una disminución, por un lado, en las costos económicos asociados a los viajes, y por otro, el lado físico se ve beneficiado, disminuyendo las fatigas causadas por los mismos.

  Para este objetivo, y como se mencionó anteriormente en esta investigación, se introduce la variable binaria $y_{i,s,t}$ igual a 1 si el equipo i viaja desde la ubicación del equipo s a la ubicación del equipo t o 0 en caso contrario. A su vez, se define la matriz D_{NxN} que representa la distancia existente entre cada par de equipos.

  Para contemplar lo antes definido se introducen las siguientes restricciones:

$$y_{i,s,t} \geq x_{s,i,k} + x_{t,i,k+1} - 1 \quad \forall k \in F : k < N-1 \quad \forall i, s, t \in E : i \neq s, t \neq s, i \tag{3.19}$$

$$y_{i,i,t} \geq \sum_{i \in E : j \neq i} (x_{i,j,k}) + x_{t,i,k+1} - 1 \quad \forall k \in F : k < N-1 \quad \forall i, t \in E : i \neq t \tag{3.20}$$

$$y_{i,t,i} \geq x_{t,i,k} + \sum_{i \in E : j \neq i} (x_{i,j,k+1}) - 1 \quad \forall k \in F : k < N-1 \quad \forall i, t \in E : i \neq t \tag{3.21}$$

  La ecuación 3.19 tiene el objetivo de que al jugar dos partidos seguidos de visitante se tenga en cuenta dicho suceso, y se cuantifique la distancia viajada desde la ubicación del equipo s a la ubicación del equipo t (am-

bos distintos al equipo i). Las dos siguientes restricciones, se encargan de que se considere la distancia viajada por un equipo hacia el estadio de otro (primero jugando de local y en la fecha siguiente de visitante) y luego el caso inverso, jugando una fecha k de visitante y luego de local en la fecha $k + 1$.

Finalmente, la función objetivo se define de la siguiente manera:

$$\text{mín } z = \sum_{i \in E} \sum_{s \in E} \sum_{t \in E: t \neq s} D_{s,t} * y_{i,s,t} \tag{3.22}$$

Dada la realidad del fútbol uruguayo, en donde en la enorme mayoría de los casos se cumple que cada equipo, luego de disputar un partido de visitante retorna a su lugar de origen, se decidió no considerar este objetivo a la hora de implementar los algoritmos que crearán los posibles fixtures. Al sumar las distancias recorridas por cada equipo, si se asume que siempre un equipo retorna a su lugar de origen luego de jugar de visitante, se obtendrá la misma distancia recorrida por cada uno de los equipos (independiente del orden en que se realice), entonces resulta sin sentido minimizar este valor. Esto se puede fundamentar en varias particularidades del campeonato uruguayo. En primer lugar, Uruguay es un país muy chico, lo que implica que las distancias a recorrer por cada equipo no sean extremadamente grandes en cada caso. Otro aspecto no menor, es que gran parte de los equipos pertenecen a Montevideo (en la edición 2021 del campeonato uruguayo de fútbol únicamente 3 participantes del total de 16 son del interior del país), entonces en casi todas las fechas no es necesaria una movilización muy grande por parte de los planteles. Por último, relacionado al punto de que cada equipo retorna a su locación después de cada partido, es importante aclarar que en el campeonato uruguayo por lo general, cada fecha se juega en los fines de semana, con lo que no resulta redituable económicamente, por ejemplo, para los equipos del interior quedarse en Montevideo esperando el siguiente partido, en caso de disputar dos fechas seguidas en esta ciudad.

3.2. Segunda versión del modelo

En la segunda iteración, para ser más consistentes con la realidad del campeonato uruguayo se tomará en consideración las dos vueltas del mismo, es decir, tanto el torneo Apertura como el Clausura. Para ello, se adaptará el modelo de la versión uno para que incorpore el formato de doble round robin, con todos los equipos enfrentándose entre sí, una vez en cada ronda.

El hecho de modelar explícitamente ambas fases (torneos Apertura y Clausura) tiene una serie de ventajas por sobre simplemente utilizar un esquema espejo o cualquiera similar. Por ejemplo, la primera ventaja es que justamente al no utilizar un esquema que determine las fechas del torneo Clausura en base a las del Apertura, existe una libertad mucho mayor, pudiéndose hasta modelar como dos torneos totalmente independientes si se quisiese. Además permitiría eventualmente, agregar restricciones que apliquen únicamente a uno de los torneos, o a un conjunto de fechas en particular, alusivas a factores geográficos, turísticos o sociales que afecten ciudades o épocas del año específicas, derivando en un fixture ampliamente más personalizable para la realidad de nuestro país, o incluso pudiendo modificarse la misma año a año. Otra ventaja se puede apreciar en la tercera restricción específica del modelo, la cual busca limitar la cantidad de partidos seguidos que equipos no top disputan contra equipos top. En este caso, es posible considerar en el conjunto de fechas seguidas, algunas pertenecientes al tramo final del torneo Apertura, y otras, a las primeras del torneo Clausura, ya que a pesar de que puedan existir varias semanas entre cada uno de los torneos, el hecho de jugar contra equipos top en este contexto puede incidir de manera psicológica en cada equipo.

En cuanto a las variables y conjuntos definidos en la primera versión, tanto la variable x como los conjuntos E y E_T permanecerán idénticos, no así F, dado que ahora se contemplarán las dos fechas del campeonato, por tanto este conjunto pasará a ser $F = \{1, ..., 30\}$

A continuación se presentarán las modificaciones pertinentes para las restricciones agregadas en la iteración anterior:

3.2.1. Restricciones generales

- Ningún equipo puede jugar contra sí mismo.
  Esta restricción se mantiene igual a la del primer modelo.

3.2.2. Restricciones round robin

- Cada equipo deberá jugar contra cada uno del resto de los equipos, tanto de local como de visitante, siendo estos dos partidos uno en cada ronda del campeonato.

$$\sum_{k \in F : k \leq \frac{|F|}{2}} (x_{i,j,k} + x_{j,i,k}) = 1 \quad \forall i,\, j \in E : i \neq j \tag{3.23}$$

$$\sum_{k \in F : k > \frac{|F|}{2}} (x_{i,j,k} + x_{j,i,k}) = 1 \quad \forall i,\, j \in E : i \neq j \tag{3.24}$$

$$\sum_{k \in F} x_{i,j,k} = 1 \quad \forall i,\, j \in E : i \neq j \tag{3.25}$$

- Cada equipo deberá jugar un partido por fecha ya sea de local o de visitante.
  Esta restricción se mantiene igual a la del primer modelo.

3.2.3. Restricciones específicas

- Peñarol y Nacional no pueden enfrentarse en las primeras M fechas, en cada una de las rondas.

$$\sum_{k \in F : k \leq M} (x_{PE\tilde{N},NAC,k} + x_{NAC,PE\tilde{N},k}) = 0 \tag{3.26}$$

$$\sum_{k \in F : k \in (N, N+M-1)} (x_{PE\tilde{N},NAC,k} + x_{NAC,PE\tilde{N},k}) = 0 \tag{3.27}$$

- Debe haber al menos P partidos entre equipos top en las últimas M fechas de cada una de las rondas.

$$\sum_{i \in E_T} \sum_{j \in E_T : j \neq i} \sum_{k \in (N-M, N)} x_{i,j,k} \geq P \tag{3.28}$$

$$\sum_{i \in E_T} \sum_{j \in E_T : j \neq i} \sum_{k > |F| - M} x_{i,j,k} \geq P \tag{3.29}$$

- Equipos no top no pueden jugar más de M fechas seguidas contra equipos top.
  Esta restricción se mantiene igual a la del primer modelo debido a que se consideran como fechas seguidas, el conjunto de fechas consecutivas que contenga tanto fechas de la primera ronda como de la segunda.

- Esta restricción intenta balancear la cantidad de partidos a jugar contra equipos top en cada mitad del campeonato para todos los equipos.

$$\sum_{k \in F : k \leq \frac{|F|}{2}} \sum_{j \in E_T : j \neq i} x_{i,j,k} \geq \frac{|E_T|}{5} + 1 \quad \forall i \in E \tag{3.30}$$

$$\sum_{k \in F : k > \frac{|F|}{2}} \sum_{j \in E_T : j \neq i} x_{i,j,k} \geq \frac{|E_T|}{5} + 1 \quad \forall i \in E \tag{3.31}$$

- No puede haber más de un partido entre equipos top en la misma fecha.
  Esta restricción se mantiene igual a la del primer modelo.

3.2.4. Posibles funciones objetivo

- **Minimizar cantidad de breaks**
  Siguiendo con el mismo razonamiento de la primera versión, la idea en este caso va a ser minimizar la cantidad de break tanto para el torneo Apertura como para el torneo Clausura. Se hace la salvedad que si un equipo juega de local en la última fecha del Apertura y luego repite la localía en la primera fecha del Clausura, esto no se considerará un break, dado que por lo general existe una ventana de tiempo de varias semanas entre cada uno de los campeonatos, con lo que se pierde la noción de break en este caso de borde y particular.

  Por lo tanto, la formulación matemática para este objetivo pasa a ser la siguiente:

$$\sum_{j \in E: j \neq i} (x_{i,j,k} + x_{i,j,k+1}) \leq 1 + v_{i,k} \quad \forall k \in (1, \frac{|F|}{2} - 1),\ \forall i \in E \tag{3.32}$$

$$\sum_{j \in E: j \neq i} (x_{i,j,k} + x_{i,j,k+1}) \leq 1 + v_{i,k} \quad \forall k \in (\frac{|F|}{2} + 1, |F| - 1),\ \forall i \in E \tag{3.33}$$

$$\sum_{j \in E: j \neq i} x_{i,j,k} \geq v_{i,k} \quad \forall k \in F \backslash \{30\},\ \forall i \in E \tag{3.34}$$

$$\sum_{j \in E: j \neq i} x_{i,j,k+1} \geq v_{i,k} \quad \forall k \in F \backslash \{30\},\ \forall i \in E \tag{3.35}$$

$$v_{i,\frac{|F|}{2}} = 0 \quad \forall i \in E \tag{3.36}$$

Como se puede apreciar más arriba, las ecuaciones 3.32 y 3.33 derivan de la ecuación 3.10. Esto va de la mano con la nueva restricción de la ecuación 3.36, que imposibilita que haya un break en la fecha quince (última fecha del torneo Apertura). Si se mantuviera la restricción de la ecuación 3.10 de la primera versión incambiada (salvo por el conjunto de fechas), se estaría impidiendo que el mismo equipo juegue de local en la última fecha del Apertura y en la primera del Clausura, un hecho que no es de interés restringir.

Por otra parte, las restricciones de las ecuaciones 3.34 y 3.35 cumplen idéntica función que sus análogas (ecuaciones 3.11 y 3.12) de la primera versión.

Para finalizar, dado que en la fecha quince se fuerza a que la variable que contabiliza los breaks valga 0, la formulación de la función objetivo también es análoga a la de la versión anterior.

Función objetivo

$$\text{mín } z = \sum_{i \in E} \sum_{k \in F \setminus \{30\}} v_{i,k} \tag{3.37}$$

- **Balancear el efecto arrastre entre todos los equipos**
  En este caso, y debido a que en general ambas fases del torneo se encuentran separadas por al menos un par de semanas, se decidió realizar una partición entre ambos torneos cortos, a la hora de contabilizar las fechas en que cada equipo se ve beneficiado por el efecto arrastre, ya que se considera que no existe una ventaja deportiva real en estos casos. Dicho de otra forma, se evitará considerar el caso borde en donde un equipo se enfrenta a un rival en la primera fecha del torneo Clausura, que a su vez disputó en la última fecha del Apertura un partido contra un equipo top.

  A efectos prácticos, la única modificación con el modelado de la versión anterior, sería no considerar el k correspondiente a la primera fecha del torneo Clausura. Por tanto, la ecuación queda como se puede apreciar a continuación:

$$\sum_{k \in F : k \neq 1, k \neq \frac{|F|}{2}+1} a_{i,k} = a_i \quad \forall i \in E \tag{3.38}$$

- **Minimizar distancias viajadas**
  Para esta versión del modelado, se mantiene la misma idea con respecto a la minimización de las distancias viajadas por cada equipo, considerando ahora las dos vueltas del campeonato. Dado que no es de interés considerar la distancia viajada entre la última fecha del torneo Apertura y la primera fecha del Clausura, ya que ambos torneos pueden estar separados entre sí por varias semanas, en lugar de considerar el conjunto total de fechas para esta formulación, se realizará una partición en las mismas. En dicha partición se considera las fechas del torneo Apertura por un lado, y las del Clausura por otro, para tener en cuenta el caso borde mencionado.

  Por tanto, las ecuaciones que derivan de lo anterior para las restricciones

presentadas en la versión uno son las siguientes:

$$y_{i,s,t} \geq x_{s,i,k} + x_{t,i,k+1} - 1 \quad \forall k \in F : k < N-1 \quad \forall i, s, t \in E : i \neq s, t \neq s, i \tag{3.39}$$

$$y_{i,s,t} \geq x_{s,i,k} + x_{t,i,k+1} - 1 \quad \forall k \in (N+1, |F|-1) \quad \forall i, s, t \in E : i \neq s, t \neq s, i \tag{3.40}$$

$$y_{i,i,t} \geq \sum_{i \in E : j \neq i} (x_{i,j,k}) + x_{t,i,k+1} - 1 \quad \forall k \in F : k < N-1 \quad \forall i, t \in E : i \neq t \tag{3.41}$$

$$y_{i,i,t} \geq \sum_{i \in E : j \neq i} (x_{i,j,k}) + x_{t,i,k+1} - 1 \quad \forall k \in (N+1, |F|-1) \quad \forall i, t \in E : i \neq t \tag{3.42}$$

$$y_{i,i,t} \geq \sum_{i \in E : j \neq i} (x_{i,j,k}) + x_{t,i,k+1} - 1 \quad \forall k \in F : k < N-1 \quad \forall i, t \in E : i \neq t \tag{3.43}$$

$$y_{i,i,t} \geq \sum_{i \in E : j \neq i} (x_{i,j,k}) + x_{t,i,k+1} - 1 \quad \forall k \in (N+1, |F|-1) \quad \forall i, t \in E : i \neq t \tag{3.44}$$

Es importante destacar que la función objetivo permanece idéntica, ya que el caso de borde inicial no la afecta.

3.3. Tercera versión del modelo

En esta tercera y última iteración, se agregará un nuevo nivel de complejidad en el modelo. En esta etapa para cada fecha se consideran los días que la conforman, es decir, se buscará asignar los partidos a un día particular,

sin dejar de cumplir las restricciones anteriores. Para poder implementar este nuevo modelo, sea $D = \{1, 2, 3\}$ el conjunto de días en una fecha, se agrega una nueva variable $d \in D$, siendo $x_{i,j,k,d}$ igual a 1 si el equipo i juega de local contra el equipo j en el día d de la fecha k, y 0 en otro caso.

Ahora se mostrarán las restricciones pertenecientes a esta nueva versión así como las modificaciones realizadas en las pertenecientes a la versión de modelo anterior:

3.3.1. Restricciones generales

- Ningún equipo puede jugar contra sí mismo.

$$\sum_{d \in D} x_{i,i,k,d} = 0 \quad \forall i \in E,\ \forall k \in F \tag{3.45}$$

3.3.2. Restricciones round robin

- Cada equipo deberá jugar contra cada uno del resto de los equipos, tanto de local como de visitante, siendo estos dos partidos uno en cada ronda del campeonato.

$$\sum_{k \in F: k \leq 15} \sum_{d \in D} (x_{i,j,k,d} + x_{j,i,k,d}) = 1 \quad \forall i,\ j \in E : i \neq j \tag{3.46}$$

$$\sum_{k \in F: k > 15} \sum_{d \in D} (x_{i,j,k,d} + x_{j,i,k,d}) = 1 \quad \forall i,\ j \in E : i \neq j \tag{3.47}$$

$$\sum_{k \in F} \sum_{d \in D} x_{i,j,k,d} = 1 \quad \forall i,\ j \in E : i \neq j \tag{3.48}$$

- Cada equipo deberá jugar un partido por fecha ya sea de local o de visitante.

$$\sum_{j \in E: j \neq i} \sum_{d \in D} (x_{i,j,k,d} + x_{j,i,k,d}) = 1 \quad \forall i \in E,\ \forall k \in F \tag{3.49}$$

3.3.3. Restricciones específicas

- Peñarol y Nacional no pueden enfrentarse en las primeras M fechas, en cada una de las rondas.

$$\sum_{k\in F:k\leq M}\sum_{d\in D}(x_{PEÑ,NAC,k,d}+x_{NAC,PEÑ,k,d})=0 \tag{3.50}$$

$$\sum_{k\in F:k\in(N,N+M-1)}\sum_{d\in D}(x_{PEÑ,NAC,k,d}+x_{NAC,PEÑ,k,d})=0 \tag{3.51}$$

- Debe haber al menos P partidos entre equipos top en las últimas M fechas de cada una de las rondas.

$$\sum_{i\in E_T}\sum_{j\in E_T:j\neq i}\sum_{k\in(N-M,N)}\sum_{d\in D}x_{i,j,k,d}\geq P \tag{3.52}$$

$$\sum_{i\in E_T}\sum_{j\in E_T:j\neq i}\sum_{k>|F|-M}\sum_{d\in D}x_{i,j,k,d}\geq P \tag{3.53}$$

- Equipos no top no pueden jugar más de M fechas seguidas contra equipos top.

$$\sum_{j\in E_T}\sum_{z=0}^{M}\sum_{d\in D}(x_{i,j,k+z,d}+x_{j,i,k+z,d})\leq M \quad \forall k\in F:k\leq|F|-M,\forall i\in E\backslash E_T \tag{3.54}$$

- Esta restricción intenta balancear la cantidad de partidos a jugar contra equipos top en cada mitad del campeonato para todos los equipos.

$$\sum_{k\in F:k\leq\frac{|F|}{2}}\sum_{j\in E_T:j\neq i}\sum_{d\in D}x_{i,j,k,d}\geq\frac{|E_T|}{5}+1 \quad \forall i\in E \tag{3.55}$$

$$\sum_{k\in F:k>\frac{|F|}{2}}\sum_{j\in E_T:j\neq i}\sum_{d\in D}x_{i,j,k,d}\geq\frac{|E_T|}{5}+1 \quad \forall i\in E \tag{3.56}$$

- No puede haber más de un partido entre equipos top en la misma fecha.

$$\sum_{i\in E_T}\sum_{j\in E_T:j\neq i}\sum_{d\in D}x_{i,j,k,d}\leq 1 \quad \forall k\in F \tag{3.57}$$

- Deben jugarse al menos N partidos en cada día de cada fecha.

$$\sum_{i \in E} \sum_{j \in E: j \neq i} x_{i,j,k,d} \geq N \quad \forall k \in F,\ \forall d \in D \tag{3.58}$$

- Peñarol y Nacional no pueden jugar el mismo día.

$$\sum_{j \in E \setminus \{PE\tilde{N}, NAC\}} \sum_{d \in D} (x_{PE\tilde{N},j,k,d} + x_{j,PE\tilde{N},k,d} + x_{NAC,j,k,d} + x_{j,NAC,k,d}) \leq 1$$
$$\forall k \in F \tag{3.59}$$

- Equipos del interior no pueden tener más de N fechas seguidas en Montevideo. Para esta restricción, se define el conjunto $E_I \subset E$, correspondiente únicamente a los equipos que ofician de local en ciudades del interior del país.

$$\sum_{j \in E \setminus E_I} \sum_{z=0}^{N-1} \sum_{d \in D} x_{i,j,k+z,d} \leq N \quad \forall i \in E_I,\ \forall k \in F : k < |F| - N \tag{3.60}$$

- En la fecha que se juegue el partido clásico, los demás equipos top deben jugar en los restantes días de esa fecha.

$$\sum_{j \in E \setminus \{PE\tilde{N}, NAC\}: j \neq i} (x_{i,j,k,d} + x_{j,i,k,d}) * (x_{PE\tilde{N},NAC,k,d} + x_{NAC,PE\tilde{N},k,d}) = 0$$
$$\forall k \in F, \forall d \in D, \forall i \in E_T \setminus \{PE\tilde{N}, NAC\} \tag{3.61}$$

3.3.4. Posibles funciones objetivo

- **Minimizar cantidad de breaks**
  El concepto de break es algo a nivel de fechas, por lo que la consideración de los días de cada una de ellas, no afecta directamente a este objetivo, por lo tanto las ecuaciones de la minimización de los breaks en la versión tres son las siguientes:

$$\sum_{j \in E: j \neq i} \sum_{d \in D} (x_{i,j,k,d} + x_{i,j,k+1,d}) \leq 1 + v_{i,k} \quad \forall k \in (1, \frac{|F|}{2} - 1), \forall i \in E \tag{3.62}$$

$$\sum_{j \in E: j \neq i} \sum_{d \in D} (x_{i,j,k,d} + x_{i,j,k+1,d}) \leq 1 + v_{i,k} \quad \forall k \in (\frac{|F|}{2} + 1, |F| - 1), \forall i \in E \tag{3.63}$$

$$\sum_{j \in E: j \neq i} \sum_{d \in D} x_{i,j,k,d} \geq v_{i,k} \quad \forall k \in F \backslash \{30\}, \forall i \in E \tag{3.64}$$

$$\sum_{j \in E: j \neq i} \sum_{d \in D} x_{i,j,k+1,d} \geq v_{i,k} \quad \forall k \in F \backslash \{30\}, \forall i \in E \tag{3.65}$$

Tanto la restricción de la ecuación 3.36 como la formulación del objetivo, al ser independientes de los días de cada fecha permanecen idénticos.

- **Balancear el efecto arrastre entre todos los equipos**
  En este caso, el efecto arrastre tiene sentido considerarlo como un aspecto que incide fecha a fecha, sin importar en qué día de la misma se dispute cada encuentro. Por lo tanto, el objetivo será el mismo de la versión anterior, permaneciendo todas sus ecuaciones inalteradas.

- **Minimizar distancias viajadas**
  Al igual que el caso anterior, este objetivo se encuentra relacionado a disminuir en la mayor medida posible, lo que cada equipo viaja para jugar de una fecha a la siguiente. Por lo tanto, el modelado de esta versión del problema, que agrega la contemplación de los días pertenecientes a cada una de las fechas, termina siendo irrelevante para este objetivo. Esto conlleva a que las ecuaciones de la versión anterior también permanezcan inalteradas, con la salvedad que ahora se considerarán todos los k y los d en lugar de los k, ignorando en el día que se juegue el partido.

- **Balanceo de los partidos que participan equipos top entre los días de cada fecha**
  En este objetivo se intenta balancear en cada fecha los partidos en los cuales participan equipos top, entre los días en que se divida cada una de ellas. Con esto se busca que los espectadores tengan la posibilidad de ver

la mayor cantidad de partidos posible de los más interesantes en cada fecha. Esto genera fixture muy atractivo, y obteniendo a su vez como consecuencia, una mayor recaudación en el torneo.

Se definen dos variables auxiliares, d_k que representa la cantidad de partidos que participan equipos top jugados en la fecha k, y e_k que significa el promedio de partidos en los que participan equipos top en los días de cada fecha.

$$\sum_{i \in E_T} \sum_{j \in E: j \neq i} \sum_{d \in D} (x_{i,j,k,d} + x_{j,i,k,d}) = d_k \quad \forall k \in F \tag{3.66}$$

$$[\sum_{i \in E_T} \sum_{j \in E: j \neq i} \sum_{d \in D} (x_{i,j,k,d} + x_{j,i,k,d})]/|D| = e_k \quad \forall k \in F \tag{3.67}$$

Por lo tanto minimizar la distancia entre estas dos variables en cada uno de los días de cada fecha, consiste en minimizar la función objetivo:

$$\text{mín}\, z = \sum_{k \in F} \sum_{d \in D} d_k - e_k \tag{3.68}$$

Capítulo 4

Algoritmo multi-objetivo para el diseño de un fixture

Uno de los puntos importantes de este proyecto, consta en abordar el problema planteado siguiendo un enfoque multi-objetivo, que permita optimizar múltiples objetivos al mismo tiempo. Este es un problema computacionalmente muy complejo, por lo que para poder resolverlo es necesario utilizar técnicas no exactas. En esta sección se dará un paneo general sobre los problemas multi-objetivo y definiciones importantes que serán utilizadas posteriormente en el análisis de resultados. Finalmente, se explica cómo se diseñó el algoritmo genético multi-objetivo para la resolución de este problema, explicando sus respectivos operadores, y las decisiones tomadas para llegar a este enfoque.

4.1. Optimización multi-objetivo

Un problema de optimización multi-objetivo, como su nombre lo indica, busca minimizar o maximizar al mismo tiempo una serie de objetivos, sujeto a una serie de restricciones que deben ser cumplidas para considerar a una solución como factible. Entonces, este tipo de problemas se puede definir de la siguiente manera (Deb, 2001):

$$\left.\begin{array}{rll} \text{Minimizar/Maximizar} & f_m(\text{x}), & m = 1, 2, \ldots, M \\ \text{sujeto a} & g_j(\text{x}) \geq 0, & j = 1, 2, \ldots, \text{J} \\ & h_k(\text{x}) = 0, & k = 1, 2, \ldots, \text{K} \\ & x_i^{(\text{L})} \leq x_i \leq x_i^{(\text{L})}, & i = 1, 2, \ldots, n \end{array}\right\} \quad (4.1)$$

En donde x es un vector solución de n variables de decisión: $x = (x_1, x_2, \ldots, x_n)^T$. Además, el último conjunto de restricciones se encarga de limitar tanto inferior como superiormente cada variable de decisión de x, generando el denominado espacio de decisión D. Por otro lado, se encuentran las funciones g_i y h_k, correspondientes a inecuaciones e igualdades con 0. Es importante destacar que las restricciones de la forma *menor o igual a* pueden ser transformadas a un formato *mayor o igual a*, simplemente multiplicándolas por -1.

Una solución x, se denomina factible, si y solo si cumple con las $J + K$ restricciones y con las $2N$ cotas inferiores y superiores, y no factible en caso contrario.

Otro aspecto a tener en cuenta para esta clase de problemas, es que, a diferencia de los problemas mono-objetivo, en este caso se genera un espacio de soluciones multi-dimensional, además del espacio de decisión ya mencionado anteriormente. A este nuevo espacio se le denomina espacio objetivo (Z). Mapeando esto con la formulación anterior, se deriva que dada una solución x en el espacio de decisión, va a existir un punto en el espacio objetivo, tal que $f(x) = z = (z_1, z_2, \ldots, z_M)^T$. Es posible apreciar esta correspondencia en la *Figura 4.1*.

4.2. Dominancia

El concepto de dominancia es algo ampliamente utilizado en el contexto de los problemas y algoritmos multi-objetivo. En este tipo de problemas y algoritmos se busca discernir si una solución es mejor a otra, aún cuando en muchos casos los objetivos pueden ser contradictorios entre sí (no existiendo una solu-

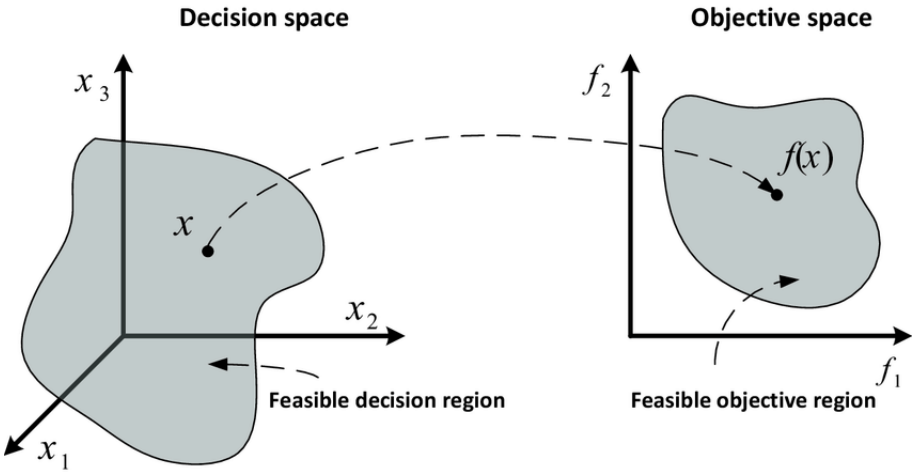

Figura 4.1: Espacio de decisión con su correspondiente espacio de objetivos. Imagen tomada de Denysiuk, 2014.

ción factible que sea óptima para todos los objetivos simultáneamente).

Asumiendo M objetivos, se dice que x_1 domina a x_2, ó $x_1 \preceq x_2$, si y solo si (Deb, 2001):

- La solución x_1 **no** es peor en todos los objetivos que x_2.
- La solución x_1 es **estrictamente** mejor en al menos **un** objetivo que x_2

A su vez, se define *conjunto no dominado*, como el conjunto de soluciones P' que **no** son dominadas por ninguna solución de un conjunto específico P.

En ocasiones, esta definición también se conoce como una relación de *dominancia débil*. Por otro lado, se define la *dominancia fuerte* de la siguiente forma: x_1 domina fuertemente a x_2, ó $x_1 \prec x_2$, si y solo si, x_1 es estrictamente mejor que x_2 para todos los M objetivos. Ahora es posible definir el conjunto débilmente no dominado, como el conjunto de soluciones $P' \subseteq P$, que **no** son fuertemente dominadas por ninguna otra solución de P.

4.3. Frente de Pareto

De la mano del concepto de dominancia ya presentado, es importante destacar algunas definiciones vinculadas a la optimalidad de *Pareto*.

Se dice que una solución x es Pareto óptima para un determinado conjunto S, si y solo si, **no** existe una solución x' que la domine. El conjunto S, si no es explícitamente aclarado, es considerado como todo el espacio de decisión existente. En otras palabras, una solución es óptima si no existe ninguna que la iguale y/o supere en todos los objetivos simultáneamente.

Por otra parte, se define el conjunto óptimo de Pareto $\mathcal{P}^*$, correspondiente al conjunto de las soluciones no dominadas, o Pareto óptimas del espacio de decisión. A su vez, si se mapean estas soluciones con sus respectivos vectores de valores objetivo, se obtiene el denominado Frente de Pareto ($\mathcal{PF}^*$), que luego suele ser utilizado para graficar los resultados obtenidos por el algoritmo multi-objetivo bajo el espacio de objetivos.

4.4. Hipervolumen

El hipervolumen es una métrica utilizada con el propósito de comparar tanto la convergencia de un conjunto de soluciones hacia el frente de Pareto, como también su diversidad y variedad. Un valor más grande del hipervolumen refleja una mejor convergencia al frente de Pareto del conjunto de soluciones.

Se considera Q, como la región rayada de la *Figura 4.2* en el espacio de objetivos. El objetivo de esta métrica es calcular el volumen de dicha zona, comprendida entre los puntos que forman el frente de Pareto y un punto de referencia W. Este punto se puede considerar como un vector con los peores valores posibles para cada objetivo. Matemáticamente, es posible considerar para cada solución $i \in Q$, un hipercubo v_i construido con el punto de referencia y dicha solución como las esquinas de la diagonal del mismo. Luego, se efectúa una unión de todos los hipercubos para obtener el hipervolumen total del conjunto de soluciones actual, presentado a continuación (Deb, 2001):

$$\text{HV} = \text{ volume } \left(\cup_{i=1}^{|Q|} v_i \right) \tag{4.2}$$

En la figura anterior, se muestra un ejemplo con dos funciones objetivos que se busca minimizar, y un frente de Pareto con cinco soluciones, por lo que el cálculo del hipervolumen se puede explicitar de la siguiente manera, con $W = (W_1, W_2)$:

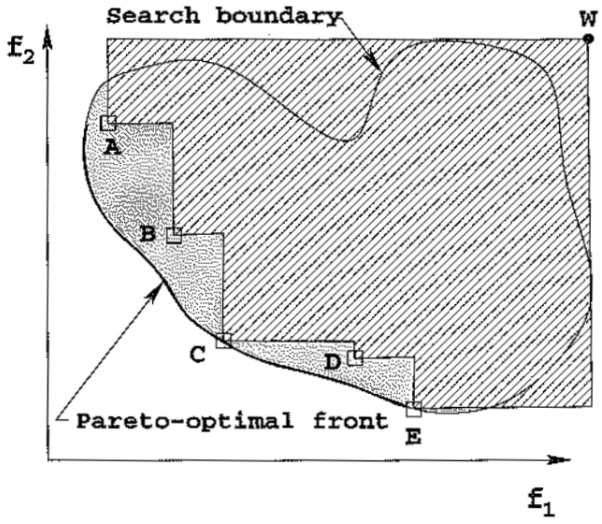

Figura 4.2: Hipervolumen con respecto a las soluciones no dominadas. Imagen tomada de Deb, 2001.

$$\begin{aligned} \text{HV} = (W_1 - f_1(A)) * (W_2 - f_2(A)) \ + \ (f_1(A) - f_1(B)) * (W_2 - f_2(B)) \\ + \ (f_1(B) - f_1(C)) * (W_2 - f_2(C)) \ + \ (f_1(C) - f_1(D)) * (W_2 - f_2(D)) \\ + \ (f_1(D) - f_1(E)) * (W_2 - f_2(E)) \end{aligned} \quad (4.3)$$

4.5. Algoritmos evolutivos

Los problemas de optimización multi-objetivo pueden ser transformados en problemas mono-objetivo, a través de una suma ponderada de los objetivos con un peso específico para cada uno de ellos. Esto tiene la desventaja de que a priori es necesario conocer el peso que debe dársele a cada uno de los objetivos. En caso contrario, existen diversos métodos para poder resolver estos problemas evitando dicha transformación. Un ejemplo, que desde hace tiempo se encuentra tomando un gran protagonismo como estrategia elegida a la hora de resolver un problema de este tipo, son los Algoritmos Evolutivos (Goldberg, 2006). Los AEs se desarrollan en torno a los principios de la evolución natural de las especies.

Una de las cualidades particulares y especiales de estos algoritmos, es el hecho de que a medida que el algoritmo va evolucionando, se van encontrando

y manteniendo múltiples soluciones para una única ejecución. De esta forma, es posible tener un abanico de soluciones que permitan, de acuerdo a los requerimientos y especificaciones de cada problema en particular, encontrar las soluciones que mejor se adecúen y cumplan *mejor* los objetivos. El mecanismo de funcionamiento de los AEs hace posible encontrar soluciones en donde no es posible discernir y detectar cuál es mejor, debido a que una solución x_1 puede ser mejor que una solución x_2 en un objetivo pero que ocurra lo contrario para otra función objetivo.

Dentro de los algoritmos evolutivos, es posible encontrar distintas opciones, donde la más utilizada, que se tomó como base para el desarrollo de la solución en este proyecto, son los algoritmos genéticos. A continuación se describirá detalladamente el funcionamiento de este tipo de algoritmos, presentando sus principales componentes. El esqueleto básico de este tipo de algoritmos es el siguiente:

Algoritmo 1: Pseudo-código Algoritmo Genético

```
Generar población inicial
Calculo de fitness para esa población
repetir
    Selección
    Cruzamiento
    Mutación
    Cálculo de fitness para los nuevos individuos
    Reemplazo poblacional
hasta que La población converja / se alcance la cantidad máxima de
  iteraciones;
```

La idea de este algoritmo es, como se mencionó anteriormente, utilizar la idea de la selección natural, para poder encontrar los mejores individuos que resuelvan el problema dado. Por lo tanto, en primer lugar se genera una población inicial, que no es más que un conjunto de individuos o cromosomas, cada uno de ellos es una posible solución al problema. Un individuo está compuesto por una serie de parámetros variables, denominados genes, los que en su conjunto forman el cromosoma. Es importante añadir que las poblaciones suelen tener una cantidad fija de individuos a lo largo de toda la ejecución del algoritmo.

Luego, entra en juego la función de fitness, la que permite medir cuan *bueno* es un individuo particular, en base a los valores que presenta para cada objetivo. El valor de fitness de una solución es lo que le permite a un cromosoma ser seleccionado para la reproducción y así sobrevivir en la nueva generación. Es decir, cuanto mejor sea este valor, mayor es la probabilidad de ser seleccionado.

Una vez se crea la población inicial y se le asigna el valor de fitness a cada individuo, el algoritmo entra en la etapa del loop, que se ejecuta hasta que se pueda asegurar que el conjunto de soluciones ha convergido o hasta que se haya alcanzado una cantidad máxima de generaciones. El primer paso aquí es la llamada etapa de selección, en donde, considerando la generación actual, se seleccionan los mejores cromosomas (en base a su valor de fitness), habilitándolos a que sus genes sean utilizados para la generación de la siguiente población.

La otra parte central del loop principal, son los operadores evolutivos, específicamente el cruzamiento y la mutación. En primer lugar, el operador de cruzamiento produce modificaciones más importantes, ya que involucra a pares de cromosomas, y no a uno en particular. En este caso, se seleccionan aleatoriamente dos cromosomas, y también aleatoriamente se elige un punto en la representación de cada individuo, como es posible apreciarlo en la *Figura 4.3*, en dónde se intercambian los genes desde el punto de cruzamiento indicado hasta el final. Se obtienen así dos nuevos cromosomas, que son añadidos a la población. Por otra parte, la mutación se utiliza para mantener la diversidad en la población y además para evitar una convergencia prematura. En cada iteración, algunos de los individuos son sometidos a una mutación con una probabilidad aleatoria bastante baja, la que genera que en los individuos seleccionados exista una modificación en algunos de sus genes, generando un nuevo cromosoma, el cual se añade a la población actual.

Una vez aplicados los operadores anteriores, es necesario actualizar la población para dar paso a la generación siguiente, en lo que se denomina la etapa del reemplazo. Aquí, el procedimiento a realizar es simplemente tomar la totalidad de los individuos seleccionados, más los nuevos generados a través de mutaciones y cruzamientos, y a cada uno de ellos calcularles su valor de fitness.

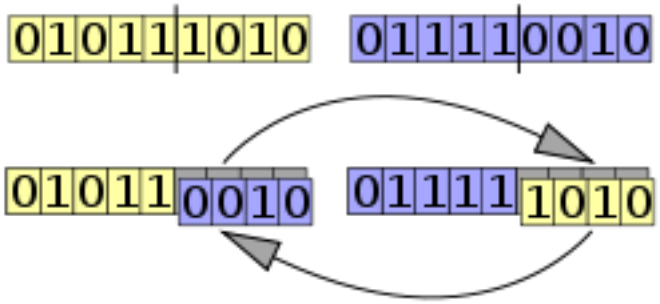

Figura 4.3: Ejemplo de cruzamiento de un punto

Luego que se hizo esto, ya es posible generar una nueva generación, aplicando algún mecanismo apropiado, como puede ser, quedarse con los mejores n individuos, o reemplazar completamente a los individuos de la generación anterior por los nuevos.

Existen dos formas de que un algoritmo genético finalice su ejecución. La primera es, definir un número fijo de generaciones, y una vez que se cumplan, retornar la mejor solución existente en esa iteración final. El otro caso, es cuando se logra una convergencia en las soluciones obtenidas, es decir, la descendencia de la generación actual presenta cambios que son prácticamente insignificantes. En ese caso, mejorar la solución encontrada por el algoritmo termina siendo casi imposible, por lo que se puede afirmar que se ha llegado al mejor valor posible, que perfectamente puede ser un óptimo local, no siendo la mejor solución posible para el problema dado.

4.5.1. Algoritmo NSGA-II

El algoritmo NSGA-II es uno de los algoritmos evolutivos más utilizados a la hora de resolver problemas multi-objetivo. Se basa en un principio elitista, esto quiere decir, que a medida que la población va evolucionando, los individuos que tienen más posibilidades de sobrevivir de una generación a otra son los mejores, es decir, los que tienen un valor de fitness más alto.

Para entender en profundidad el funcionamiento de este algoritmo se presentará a continuación una imagen que muestra visualmente el detalle en la ejecución de un algoritmo de este tipo. La imagen incluye cómo se consideran tanto la población anterior y los nuevos individuos generados mediante la

aplicación de operadores evolutivos, a la hora de seleccionar los mejores individuos. Para ello sea realiza un ordenamiento que en primer lugar separa los individuos en distintas categorías, llamados *frentes*, y luego, en caso de ser necesario, dentro de cada frente se puede realizar un nuevo ordenamiento para priorizar los mejores individuos de cada uno.

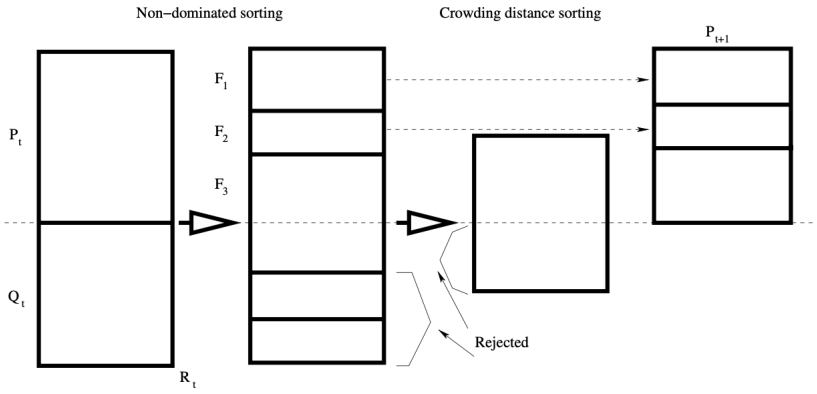

Figura 4.4: Esquema de funcionamiento del algoritmo NSGA-II. Imagen tomada de Coello et al., 2007.

Como es posible apreciar en la *Figura 4.4*, en cada iteración este algoritmo utiliza una población R_t, generada como la unión de la población padre P_t y la descendencia Q_t. Esto busca tener una mayor variedad a la hora de elegir soluciones, aumentando las posibilidades de encontrar soluciones mejores. Una vez hecho esto, se realiza una ordenación de todos los individuos en base a su nivel de dominancia, agrupándolos luego por su nivel de dominancia, en lo que se denominan *frentes*, siendo el primero de ellos, F_1, el correspondiente al conjunto de los individuos no dominados, candidatos a estar en el frente de Pareto al terminar la simulación.

Dado que para la siguiente generación del AE es necesario seleccionar N individuos, siendo N el tamaño de la población inicial, no es posible incluir todos los frentes generados en el paso anterior. Por lo que, partiendo de los mejores, en cuanto a su nivel de dominancia, se van incluyendo en el nuevo conjunto P_{t+1}, hasta que no entre ningún frente más en su totalidad. En ese momento, se pasa a realizar un segundo nivel de ordenamiento, un ordenamiento binario basado en la *crowding distance*. La crowding distance es una métrica que ayuda a obtener un estimado de la densidad de soluciones que rodean a una solución en particular. Sea una solución i, esta métrica se calcula

como el promedio de la distancia de las dos soluciones vecinas más cercanas a la solución i, considerada para cada uno de los objetivos. Por lo tanto, a la hora de elegir qué solución sobrevivirá, aún cuando pertenezcan al mismo frente, se elegirán las que posean mayor valor para esta métrica, consiguiendo así que exista un abanico de soluciones mucho más variado, con el objetivo de buscar la diversidad de soluciones en el frente.

Una aclaración importante, es que a medida que el algoritmo va iterando, es posible que las soluciones pertenecientes al frente de Pareto, es decir las que no son dominadas por ninguna otra, terminen excediendo en cantidad a N. Sin embargo, el propio mecanismo de funcionamiento del algoritmo permite seleccionar las N soluciones mejor distribuidas en el frente, ya que utilizando la crowding distance, se logra una mayor diversidad y variedad en las soluciones generadas.

4.6. Algoritmo evolutivo multi-objetivo para la elaboración de un fixture

Debido a la alta complejidad computacional que presenta la elaboración de un fixture, considerando una serie de objetivos y sujeto a un amplio conjunto de restricciones que se deben cumplir, la sola utilización de un solver exacto para este problema conlleva tiempos que son exponenciales al aumentar la cantidad de equipos o ajustando la complejidad de alguna restricción. Además, el método exacto no permite obtener el conjunto de soluciones del frente de Pareto, si no que únicamente es posible obtener de a una solución por ejecución, a través de utilizar una función objetivo que agregue en forma ponderada los distintos objetivos y realizar modificaciones en la ponderación de los pesos para cada uno de los ellos. Por ello, es de interés para la elaboración de este proyecto, diseñar un algoritmo multi-objetivo que utilice metaheurísticas para poder acercarse lo más posible al conjunto óptimo de Pareto, y a su vez teniendo en cuenta la totalidad de los objetivos planteados.

Luego de evaluar diversas posibilidades a la hora de elegir qué tipo de algoritmo utilizar, se optó por desarrollar un algoritmo evolutivo, más precisamente

un algoritmo genético de tipo NSGA-II, como se presentó en la sección anterior. Las razones para elegir este algoritmo se basan en la experiencia anterior del equipo utilizando algoritmos genéticos, y la gran efectividad mostrada en trabajos previos por el algoritmo NSGA-II (Deb, 2001) a la hora de trabajar con múltiples objetivos simultáneamente.

El esquema de la solución desarrollada está basado en el *método del polígono* (Dinitz et al., 2006), un método que utiliza propiedades matemáticas de los polígonos para desarrollar un fixture factible que cumpla con todos los requisitos de un campeonato *round robin*. En las siguientes dos secciones se explicará detalladamente el funcionamiento del método del polígono, ilustrado en base a un ejemplo para una cantidad acotada de equipos, y luego se entrará en detalle en la mutación específica que se diseñó para este problema, siendo este uno de los operadores básicos de un algoritmo genético clásico. Por otra parte, es necesario aclarar que no se implementó ningún operador de cruzamiento. Esto se debe a que al tener individuos tan particulares, como se verá en la siguiente sección, un posible operador de cruzamiento impactaría enormemente en la estructura de la solución. Esto derivaría en un fixture totalmente diferente al anterior, siendo contraproducente a la hora de facilitar la convergencia del algoritmo.

4.6.1. Método del polígono

La utilización de este método permite la creación de un fixture factible, en donde todos los equipos se enfrentan una única vez contra los demás, además jugando un único partido por cada fecha. Teniendo en cuenta que se va a trabajar con un algoritmo genético, en particular un algoritmo NSGA-II, este enfoque asegura la factibilidad en la totalidad de las soluciones generadas. Esto evita lidiar con la corrección o el descarte de soluciones no factibles en cada iteración, lo que agregaría complejidad y degradaría la performance del algoritmo. A continuación se detallará el proceso de creación de un fixture factible para un caso de ejemplo de ocho equipos.

En primer lugar, se dibuja un polígono con $N - 1$ vértices, siendo N la cantidad de equipos participantes en el torneo (ocho en el caso de ejemplo), y un punto central extra, que sumado a los vértices anteriores representarán

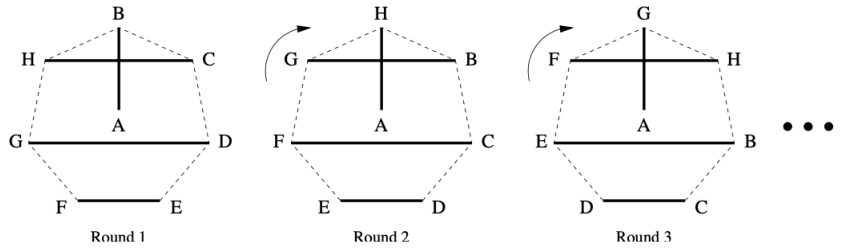

Figura 4.5: Ejemplo utilizando el método del polígono para ocho equipos. Imagen tomada de Barone et al., 2006.

a los N equipos que disputarán el campeonato. Luego, se dibujan $\frac{N}{2}$ segmentos uniendo cada par de vértices, de manera que cada uno de ellos tenga un único segmento. Cada línea de éstas simboliza un partido a disputarse entre el par de equipos involucrado. Esto se ilustra en el primer polígono de la *Figura 4.5*. De esta manera, para la primera fecha, los partidos resultantes serían $(A, B), (H, C), (G, D)$ *y* (F, E).

Para generar la siguiente fecha, es necesario realizar una rotación en los vértices un lugar en sentido horario, sin incluir el vértice central que permanecerá fijo en todos los casos. Esto genera un nuevo conjunto de partidos que son los que se disputarán en la nueva fecha, como también es posible apreciarlo en la figura anterior. Repitiendo este paso sucesivamente $N - 2$ veces, se generará el fixture completo para las $N - 1$ fechas existentes.

Con el objetivo de tener una estructura que sea más manejable, se representa la solución mediante una matriz, que en lugar de almacenar los partidos que ocurren en cada fecha, se almacena la fecha en la que se disputa cada encuentro para cada par de equipos, como se muestra en la *Figura 4.6*.

Un detalle que es importante aclarar, es que hasta ahora no se han tenido en cuenta las localías en los partidos, algo que es muy importante, por ejemplo a la hora de contar la cantidad de breaks en cada campeonato. Para ello, se agrega otra matriz, análoga a la anterior, pero que en lugar de tener en cada celda la fecha en la que se disputa el partido para cada par de equipos, incluye un booleano que indica si un equipo es local contra otro (se indica con un uno). Notar que en la celda equivalente en la matriz traspuesta, el valor debe ser el

Equipos	A	B	C	D	E	F	G	H
A	–	1	7	6	5	4	3	2
B	1	–	4	7	3	6	2	5
C	7	4	–	3	6	2	5	1
D	6	7	3	–	2	5	1	4
E	5	3	6	2	–	1	4	7
F	4	6	2	5	1	–	7	3
G	3	2	5	1	4	7	–	6
H	2	5	1	4	7	3	6	–

Figura 4.6: Fecha de los partidos para cada par de equipos utilizando el método del polígono.

opuesto, ya que el otro equipo será visitante. Esto es ilustrado en la *Figura 4.7*

Equipos	A	B	C	D	E	F	G	H
A	-	1	0	0	1	1	0	1
B	0	-	1	1	0	0	1	0
C	1	0	-	0	1	0	1	0
D	1	0	1	-	0	1	0	1
E	0	1	0	1	-	0	0	1
F	0	1	1	0	1	-	1	1
G	1	0	0	1	1	0	-	0
H	0	1	1	0	0	0	1	-

Figura 4.7: Localías de los partidos para cada par de equipos.

Es importante destacar que con la estructura propuesta hasta el momento, el fixture resultante, a excepción de las localías, siempre será el mismo. Por este motivo, es fundamental incluir en la estructura la posibilidad de variar los equipos y las fechas. Para ello, se agregan dos estructuras más. La primera para hacer un mapeo lógico utilizando equipos ficticios, que permitan fácilmente rotarlos, logrando explorar un espacio de soluciones mucho mayor. Análogamente, para también ofrecer mayor variedad de soluciones, se agrega también una estructura para hacer un mapeo lógico en el orden de las fechas.

Entonces, resumiendo lo presentado anteriormente, la manera de implementar cada fixture particular, constará de cuatro estructuras, una matriz de fechas, una matriz para las localías de los partidos, y dos estructuras que rea-

lizan un mapeo de los equipos y las fechas lógicas. Dado que el operador de mutación propuesto en este trabajo únicamente puede modificar las tres últimas estructuras de datos, la matriz de fechas se mantiene fija para todos los individuos. Por lo que no es necesario incluirla como parte de la información de cada individuo, ya que agrega una redundancia que no es necesaria y solo terminaría degradando la performance del algoritmo.

4.6.2. Operador de mutación

La mutación es uno de los dos operadores básicos en un algoritmo genético. Como se ha explicado con anterioridad, una mutación es una forma de agregar mayor diversidad a una población determinada, pero afectando a un único individuo cada vez que se aplica este operador.

Al tener individuos bastante particulares, no es posible aplicar la mayoría de las mutaciones por defecto existentes en la mayoría de las librerías que brindan soporte para desarrollar un algoritmo genético. Por este motivo se debió diseñar e implementar una mutación específica para este problema y la representación utilizada. Como fue explicado en la sección anterior, cada individuo es una tupla de tres estructuras, las cuales serán sometidas a un cambio totalmente independiente y por separado, los que se detallarán a continuación:

- **Equipos lógicos:** se eligen dos equipos cualquiera de esta estructura de datos y con una probabilidad variable y baja, se invierten las ubicaciones de estos dos equipos en la lista. Es decir que el equipo A pasa a tener el fixture de B y viceversa.
- **Fechas lógicas:** análogamente al caso anterior, se define otra probabilidad independiente y con ella se efectúa un intercambio entre dos fechas lógicas, también seleccionadas aleatoriamente. Es decir que la fecha i pasa a ser la fecha j y viceversa.
- **Matriz de local-visitante:** se eligen aleatoriamente una cantidad variable de partidos (acotada superiormente), y se procede a invertir las localías. Es decir si se eligió un partido entre un equipo A contra un equipo B, en donde A era el equipo local, luego de aplicada la mutación, el partido entre estos dos equipos se disputará en el estadio del equipo B.

4.6.3. Función de fitness

Como ya se mencionó con anterioridad, una de las partes más importantes en un algoritmo genético, que a su vez es lo que más difiere de un problema particular a otro, es la función de fitness. Este es un mecanismo que permite ayudar a definir no sólo cuan buena es una posible solución, para cada uno de los objetivos existentes, si no que también permiten discernir si una solución es *mejor* que otra.

En primer lugar, el primer objetivo que se buscó optimizar es la cantidad de breaks. En este caso, se optó por desglosarlo en más de un objetivo por separado, con la idea de no simplemente minimizar la cantidad total de breaks. Se trata de que exista un balance, ya sea entre todos los equipos participantes, así como también entre los breaks de local y visitante para cada uno de ellos. Para ello, se definieron tres objetivos independientes que buscan conseguir lo mencionado con anterioridad. El primero de ellos refiere a los breaks de local, en donde lo que se busca es la minimización de la suma de los cuadrados de breaks para cada equipo, obteniendo así un fixture en donde cada equipo tenga la menor diferencia posible de breaks de local con todos los demás. Si se minimizaran únicamente los breaks podrían darse el caso, por ejemplo, de que un equipo A tenga cinco breaks y un equipo B tenga un único break. A través de la minimización del valor recién mencionado se busca obtener una solución que permita que ambos equipos, tengan una cantidad de breaks más balanceada, lográndose un mayor nivel de justicia. Análogamente, se considera como objetivo el caso de los breaks de visitante. La razón por la que se decidió separar esto en dos objetivos individuales es que al hacer esto, no solo se busca equiparar la cantidad de breaks entre todos los equipos, si no que a su vez también se equilibra para cada uno de ellos la cantidad de breaks de cada tipo, también ayudando a encontrar un fixture más parejo y justo posible. Por último, el tercer objetivo consta en la minimización de la diferencia de partidos de local y visitante para cada equipo participante del campeonato. Así, todos los equipos juegan una cantidad de partidos muy similar en su estadio, también beneficiando al equilibrio del fixture.

El otro objetivo primordial en este proyecto se trata del balanceo del efecto arrastre. Para ello, siguiendo con la misma idea que en el caso de los breaks,

se buscó minimizar la suma de los cuadrados de este valor para cada equipo, y así lograr encontrar un balance entre todos los equipos. De esta manera se evita que un conjunto acotado de equipos se vea beneficiado por el efecto arrastre por encima de los demás.

Por otra parte, como se presentó en la sección del modelado del problema, en esta realidad se consideraron una serie de restricciones que un fixture para el campeonato uruguayo debería cumplir para ser considerado válido o factible. Uno de los diversos enfoques para incorporar restricciones a un algoritmo genético, que fue el escogido en esta solución, consiste en modelar cada una de ellas como un objetivo más, pero ponderado de distinta manera que los objetivos reales presentados antes en esta misma sección. Este enfoque permite cierta flexibilidad que no existía al utilizar el solver exacto, ya que al incluir las restricciones como un objetivo, por más que el peso relativo sea superior, puede ocurrir que existan soluciones que no cumplan con la totalidad de las restricciones. Sin embargo, se consideró que puede ser de utilidad tener en cuenta este conjunto particular de soluciones, ya que puede ser preferible violar levemente alguna de las restricciones pero obtener buenos valores en los breaks y efecto arrastre, a que directamente descartar esas soluciones.

Resumiendo lo presentado en esta sección, se puede concluir que más allá de cada aspecto particular considerado a la hora de crear el *mejor* fixture, lo más importante a destacar, es que se buscó en cada objetivo y restricción incluida, el mayor nivel de justicia posible, sin beneficiar a ningún equipo por encima de otro.

Capítulo 5

Análisis experimental

En esta sección se presentará en primer lugar un estudio de los fixtures reales que se han utilizado en el último par de años en los torneos locales de fútbol y básquetbol. Luego se realizará un análisis detallado de los resultados obtenidos en todo el proceso experimental realizado sobre el algoritmo exacto y el algoritmo evolutivo multi-objetivo utilizando distintas métricas. Por último, se hará una comparación de los resultados obtenidos con los fixtures reales, mostrando como la solución implementada pudo resolver las anomalías encontradas en el estudio de los fixtures utilizados en los campeonatos locales.

Todas las implementaciones fueron realizadas en Python 3.6.8. Para la ejecución del algoritmo exacto se utilizó la librería Gurobipy 9.1.2 y para la implementación basada en algoritmos evolutivos multi-objetivo fue usada la librería Deap 1.3.1 de Python. La plataforma de ejecución de los experimentos cuenta con un procesador Intel Core i7-6700 3.40GHz de 4 núcleos y 8 hilos con una memoria RAM de 64GB sobre un sistema operativo CentOS Linux 7.

5.1. Campeonatos reales utilizados para el análisis

Si bien el proyecto está centrado en la realización y estudio de un fixture para el Campeonato Uruguayo de Fútbol, se consideró interesante analizar también los últimos campeonatos de básquetbol. Esto se debe a que existe una cierta semejanza en cuanto a la cantidad de equipos, formato de torneo, y sobre todo considerando que las restricciones y objetivos planteados en la

sección del modelado perfectamente pueden ser aplicados a este deporte. Los fixtures de los cuatro torneos analizados son: Campeonato Uruguayo de Fútbol 2020, Campeonato Uruguayo de Fútbol 2021, Liga Uruguaya de Básquetbol (LUB) 2020/21 y Liga Uruguaya de Básquetbol (LUB) 2021/22 se encuentran detallados en el Apéndice 1.

Para la totalidad de las pruebas se consideraron torneos de una vuelta (round robin), aunque es posible ampliarlo a ambas rondas del campeonato si así se quisiera en el futuro (como se vio en el Capítulo 3). Esta decisión se justifica en que existe una separación temporal de al menos un par de semanas, lo que implica que no tenga un efecto real, a nivel de cómo afecta a los equipos con respecto a los breaks y al efecto arrastre, en el conjunto total de las fechas a disputarse. Es decir, que se podrían considerar como torneos independientes siguiendo un esquema de espejo, lo que aseguraría que en la segunda vuelta también se cumplan las restricciones y se optimicen adecuadamente los objetivos buscados. Otra ventaja de seguir este enfoque, es que al haber una cantidad de partidos considerablemente menor, se generará una complejidad computacional menor.

Tabla 5.1: Análisis de fixtures reales

Torneo	Equipos	Breaks	Efecto arrastre
Liga Uruguaya de Básquetbol 2020/21	12	34	152
Liga Uruguaya de Básquetbol 2021/22	14	54	224
Campeonato Uruguayo de Fútbol 2020	16	14	644
Campeonato Uruguayo de Fútbol 2021	16	14	620

Analizando los fixtures utilizados en los últimos dos campeonatos uruguayos de fútbol, a nivel de los objetivos, se puede apreciar viendo la *Tabla 5.1* el siguiente patrón: para ambos casos se logra llegar al mínimo teórico de breaks totales, valor dado por la fórmula $N - 2$ (van't Hof et al., 2009), siendo N la cantidad de equipos. Sin embargo, esto se logra en detrimento de otros objetivos, en particular del efecto arrastre.

A la hora de hablar del efecto arrastre es necesario definir cuáles serán los equipos top de cada torneo. Para esto se utilizó un criterio basado en el desempeño de los equipos en los últimos campeonatos y la popularidad de los

mismos a lo largo de la historia. Quedan entonces definidos en cada torneo de la siguiente forma: Campeonato Uruguayo de Fútbol 2021 - Peñarol, Nacional, Wanderers, Liverpool; Campeonato Uruguayo de Fútbol 2020 - Peñarol, Nacional, Danubio y Defensor; LUB 2021/22 - Trouville, Nacional, Urunday Universitario y Aguada; LUB 2020/21 - Biguá, Olimpia, Nacional y Urunday Universitario. Para la cantidad de equipos del campeonato de fútbol (dieciséis), el mínimo teórico para este objetivo es 200. Esto se puede explicar de la siguiente manera. En primer lugar, no se debe considerar la primera fecha, ya que el efecto arrastre no tiene sentido al no haber ningún partido previo que pueda favorecer a algún equipo en particular. Luego, considerando que van a existir cuatro equipos top en el campeonato, existirán 56 partidos (cuatro por cada fecha desde la segunda hasta la final), en donde un equipo se enfrente a otro que venga de jugar contra un equipo top. Por lo tanto, la forma más equitativa de dividir estos 56 partidos entre los dieciséis equipos que conforman el campeonato, sería que ocho de ellos se vean beneficiados por el efecto arrastre en tres oportunidades, y los ocho restantes equipos, en cuatro. Elevando al cuadrado los valores de efecto arrastre de cada equipo y luego sumándolos, se obtiene el valor 200, mencionado anteriormente (para el caso de los torneos LUB 2020/21 y LUB 2021/22 análogamente se repite el proceso con la cantidad de equipos de cada torneo, obteniendo un mínimo teórico de efecto arrastre de 136 y 168, respectivamente). Como se puede ver en la *Tabla 5.1* los valores reales alcanzados para los campeonatos en 2020 y 2021 son de 644 y 620, respectivamente. Como se puede apreciar, son valores extremadamente altos, y en ambos casos se explican por el mismo motivo que será detallado a continuación.

El fixture del campeonato uruguayo estructuralmente está elaborado con una suerte de desplazamiento fecha a fecha. Esto quiere decir que se definió, para un equipo genérico i, los $N-1$ rivales con una fecha asociada a cada uno de ellos. Luego, para el segundo equipo j, se tomó este mismo ordenamiento y desplazó una fecha hacia adelante circularmente, esto quiere decir que la última fecha del equipo i pasa a ser la primera de j, y así sucesivamente para los otros equipos. Una salvedad con respecto a este último punto es que el hecho de que j enfrente a los rivales que enfrentó i en la fecha anterior no puede ocurrir para la totalidad de las fechas. Esto se debe a que, en primer lugar, deben jugar entre sí, y además por el desplazamiento ya mencionado también

pueden ser afectadas las fechas anterior y siguiente. Por ejemplo, si A juega contra B en la fecha k, es imposible que se cumpla que para las fechas k y $k+1$, B dispute su partido correspondiente contra el equipo que en la fecha anterior se enfrentó a A, debido a que para ello A debería jugar consigo mismo en la fecha $k-1$ y B también debería jugar consigo mismo en la fecha $k+1$.

Entonces, sin considerar estos tres casos de fechas particulares, y asumiendo que A es un equipo top, se cumplirá que B será beneficiado por el efecto arrastre en un alto número de partidos, dado que en la mayoría de las fechas, sus rivales habrán jugado la fecha anterior contra un equipo top. Esto es lo que sucede para los últimos fixtures del campeonato uruguayo, en donde habrán tantos equipos favorecidos por el efecto arrastre como equipos top se consideren, sin importar cuáles se elijan, ya que en esos casos la única diferencia existente será relativa a qué equipo va a ser beneficiado y cual no, pero el patrón se seguirá cumpliendo en ambos casos.

Para dar un ejemplo concreto de lo presentado recién, y basándose en el fixture del Campeonato Uruguayo de Fútbol 2021 (*Tabla A1.2*), se puede apreciar que Peñarol juega contra Villa Española, Nacional y Cerrito en las fechas ocho, nueve y diez, respectivamente, mientras que Montevideo City Torque se enfrenta a los mismos rivales, pero en las fechas nueve, diez y once, respectivamente. En este pequeño fragmento se demuestra el desplazamiento fecha a fecha utilizado para la elaboración del fixture. A su vez, este último equipo se ve beneficiado del efecto arrastre, al jugar la mayoría de los partidos contra equipos que vienen de enfrentar a Peñarol en la fecha anterior.

5.2. Análisis de resultados con algoritmo de programación entera

Para la implementación del modelo basado en programación entera se utilizó Gurobi Optimizer (Gurobi Optimization, LLC, 2022), un solver de optimización matemática muy usado y reconocido a nivel mundial por su capacidad de resolver problemas altamente complejos en tiempos relativamente bajos. Este tipo de solver recibe como entrada un modelo de programación entera,

para así poder brindar, en caso de existir, una solución exacta al problema en cuestión. De lo contrario, la ejecución finalizará indicando que no existe una solución factible para el problema ya mencionado.

La obtención de una solución exacta, dado un conjunto de restricciones y un objetivo el cual se quiere, ya sea minimizar como maximizar, fue una de las principales razones por las cuales se optó por buscar una solución al problema a través de un modelo de programación entera, ya que el mismo asegura el cumplimiento de todas las restricciones especificadas en el modelo.

A la hora de especificar las restricciones en el modelo, hay que tener en cuenta qué tan restrictivas pueden llegar a ser, ya que cuanto más restrinjan al modelo, menor va a ser la cantidad de soluciones factibles al problema, pudiendo incluso resultar en la no factibilidad del mismo. Un claro ejemplo de esto se da cuando dos restricciones se contradicen entre sí, haciendo imposible al solver que encuentre una solución que satisfaga ambas restricciones. Por ejemplo, que el clásico entre Peñarol y Nacional se juegue en la primer mitad del campeonato, junto con que todos los partidos entre equipos top se jueguen en la segunda mitad del campeonato forman un conjunto de restricciones contradictorias que resultan en la no factibilidad del problema a resolver.

Una de las mayores desventajas de los solvers de optimización matemática es que sólo soportan la optimización de un objetivo a la vez, por lo que para lidiar con problemas de más de un objetivo, se deben correr ejecuciones distintas, una para cada uno de ellos utilizando siempre las mismas restricciones. Otra alternativa podría ser hacer una nueva función objetivo como la combinación lineal ponderada de los dos objetivos en cuestión. Este enfoque implicaría asignarle a priori una cierta relevancia relativa a cada uno de los objetivos, lo cual resulta en general difícil.

Utilizando la primera versión del modelo, se realizaron ejecuciones para tres tamaños distintos de torneos (de 12, 14 y 16 equipos), dos por cada tamaño (una con el objetivo de minimizar los breaks, y otra con el efecto arrastre). La primera observación que se notó, fue que todas las ejecuciones retornaron soluciones factibles, por lo que es posible asegurar que las restricciones planteadas no son contradictorias. Se definieron dos condiciones de parada. La primera por

tiempo máximo de ejecución, con un valor de 72 horas ya que se pudo concluir que ese tiempo es lo suficientemente prudente como para resolver el problema en cuestión. La segunda refiere al valor del objetivo de la mejor solución obtenida hasta el momento, es decir, cuando se llegue a un cierto valor especificado la ejecución terminará. Para esta condición se utilizó el valor óptimo correspondiente a cada objetivo ya que no existe una solución que pueda mejorar esos valores. Dicho valor varía dependiendo del objetivo que se esté considerando. Como se dijo en la sección anterior, para el caso de los breaks, si se considera un torneo de N equipos, el valor óptimo es $N - 2$, mientras que para el efecto arrastre los valores óptimos para torneos de 12, 14 y 16 equipos son 136, 168 y 200, respectivamente.

Tabla 5.2: Minimización de breaks

Equipos	Breaks	Efecto arrastre	Tiempo de ejecución (minutos)
12	**10**	158	112
14	**12**	238	1162
16	16	252	4320*

Valores óptimos de breaks marcados en negrita
Tiempos máximos de ejecución marcados con *

Tabla 5.3: Minimización de efecto arrastre

Equipos	Breaks	Efecto arrastre	Tiempo de ejecución (minutos)
12	56	**136**	665
14	76	**168**	2603
16	102	202	4320*

Valores óptimos de efecto arrastre marcados en negrita
Tiempos máximos de ejecución marcados con *

La *Tabla 5.2* y *5.3* presentan los resultados obtenidos cuando se minimiza la cantidad de breaks y de efecto arrastre, respectivamente. Como se puede apreciar en las tablas, de las seis ejecuciones realizadas, las dos que contienen un torneo de 16 equipos fueron las únicas que no alcanzaron los valores óptimos dentro del tiempo máximo de ejecución (4320 minutos). Esto se debe a que un fixture de 16 equipos exige un poder de cómputo muy alto para poder obtener valores óptimos para el problema estudiado. Cabe destacar que aunque no se pudo llegar al valor óptimo, la convergencia estuvo cerca de ser alcanzada, ya

que se consiguieron valores muy cercanos a los esperados, llegando a 16 breaks (*Tabla A2.1*) y 202 de efecto arrastre (*Tabla A2.2*), siendo 14 y 200 los valores óptimos, respectivamente.

Si se hace énfasis en los tiempos de ejecución, estos aumentan de forma exponencial a medida que aumenta la cantidad de equipos del torneo. Esto permite concluir que este tipo de modelos no escala para torneos de grandes cantidades de equipos, ya que requiere una cantidad considerablemente grande de poder de cómputo por un tiempo prolongado.

En lo que respecta a los valores obtenidos para cada uno de los objetivos a minimizar, se puede ver que en cuatro de las seis ejecuciones se pudo alcanzar el valor óptimo para el objetivo considerado. Ahora, si se tiene en cuenta el objetivo que no fue minimizado, es decir en el caso de minimización de breaks, el valor del efecto arrastre y viceversa, se puede notar que los resultados obtenidos están bastante alejados del óptimo. Esto era predecible ya que no se están optimizando los dos objetivos en simultáneo.

Si se analizan las soluciones obtenidas y se las compara con los fixtures reales mencionados en secciones anteriores, primero se puede notar que en los torneos de 12 y 14 equipos correspondientes a la LUB 2020/21 y a la LUB 2021/22 se obtuvieron soluciones con mejores valores de breaks (*Tabla A2.3* y *A2.5*) y efecto arrastre (*Tabla A2.4* y *A2.6*) individualmente. Es decir, que a pesar de haber conseguido valores significativamente mejores, en ninguna de las soluciones se obtuvieron mejores valores en ambos objetivos en simultáneo que en los utilizados en la Liga Uruguaya de Básquetbol. Vale la pena mencionar que en las soluciones brindadas por la ejecución que buscaba minimizar los breaks, en ambos tamaños de torneo se logró llegar a un valor muy cercano al valor de los fixtures reales en el objetivo sin minimizar (158 vs 152 en 2020/21 y 238 vs 224 en 2021/22), por lo que podrían ser consideradas mejores soluciones que las utilizadas en años anteriores (ya que mejoran notablemente el objetivo de minimizar los breaks).

A la hora de comparar los resultados obtenidos en torneos de 16 equipos correspondientes al Campeonato Uruguayo de Fútbol 2020 y 2021, la primer gran observación refiere a la gran diferencia en cuanto al valor del efecto arras-

tre entre los torneos del Campeonato Uruguayo y la solución obtenida en la ejecución de minimización del objetivo en cuestión. Pese a que se pudo reducir de modo importante este objetivo, la cantidad de breaks resultante en dicha ejecución no fue buena, ya que se consiguieron valores muy alejados del óptimo.

Poniendo foco en el efecto arrastre (partidos en que se beneficia un equipo por jugar contra otro el cual viene de jugar contra un equipo top) se puede ver que se logró una gran reducción de este objetivo. Por ejemplo, en el caso del Campeonato Uruguayo de Fútbol 2020 (*Tabla A1.1*) los dos equipos con mayor valor del objetivo en el torneo (Danubio y River) se benefician en trece fechas de efecto arrastre, mientras que la mayoría de los equipos del fixture obtenido llegan a beneficiarse en solo cuatro fechas, siendo este el valor máximo de efecto arrastre por equipo. Esto nos garantiza un campeonato más justo ya que gracias al reducido valor de dicho objetivo no existe gran diferencia entre los valores de efecto arrastre de cada equipo. Como se dijo previamente, en la ejecución de minimización de breaks no se pudo llegar al valor óptimo aunque si a un valor muy cercano, y de igual forma el efecto arrastre resultante fue ampliamente mejor a los fixtures utilizados por la AUF, siendo posiblemente esta solución mejor considerada que las utilizadas anteriormente.

A partir del análisis realizado en esta sección, se puede concluir que con este tipo de aproximación al problema no es posible obtener resultados que satisfagan a todos los objetivos a la misma vez, ya que cuando se minimiza un objetivo los valores resultantes del otro objetivo están cada vez más alejados del valor óptimo. Por lo que en caso de querer optimizar ambos objetivos al mismo tiempo, se debe buscar un enfoque distinto. Es aquí donde entran en juego los algoritmos evolutivos multi-objetivo.

5.3. Análisis experimental del algoritmo NSGA-II

El análisis experimental del algoritmo NSGA-II desarrollado consiste, en primer lugar, en un estudio del impacto del tamaño de población utilizado, que permitió definir el valor que será usado en el resto del análisis. Luego, se analizó cómo afecta la cantidad de iteraciones utilizadas en cada ejecución a las soluciones generadas. Siguiendo con el análisis del algoritmo NSGA-II, se incluyó un análisis de convergencia de las soluciones generadas por medio de un estudio del frente de Pareto, a medida que las generaciones van evolucionando durante la ejecución. Para concluir, se hizo un análisis exhaustivo de los resultados obtenidos en las diversas ejecuciones realizadas, y a su vez comparándolos con los resultados obtenidos por medio de los algoritmos implementados de programación entera, así como también con los fixtures reales utilizados en el último par de años para el Campeonato Uruguayo de Fútbol y para la Liga Uruguaya de Básquetbol.

5.3.1. Análisis del efecto del tamaño de población

El objetivo en esta sección es analizar el rendimiento del algoritmo en base a distintos tamaños de población. Un tamaño de población muy bajo implica necesariamente que el algoritmo demora menos tiempo en terminar, ya que evalúa una menor cantidad de soluciones, pero a su vez la calidad de las soluciones no es buena y se pueden encontrar menos individuos en el frente de Pareto. Por el contrario, un tamaño de población muy alto implica que el algoritmo obtenga soluciones que son mejores en cuanto a la calidad, y también se podrán encontrar más individuos en el frente de Pareto, pero a costo de un tiempo de ejecución considerablemente más alto.

Para analizar el efecto del tamaño de la población en los resultados, se realizaron diez ejecuciones para diez tamaños de población distintos, de 10000 generaciones cada una de ellas, y se calculó el promedio del hipervolumen para cada uno de los tamaños considerados (de 100 en 100 hasta el tamaño final de 1000). Para calcular el hipervolumen, se consideraron los peores y mejores valores teóricamente posibles de cada objetivo y se utilizaron para normalizar

los valores de los objetivos de cada individuo del frente de Pareto entre 0 y 1. Luego, se tomó como referencia el punto (1, 1, 1) y se calculó el hipervolumen. Cabe destacar que como los peores valores posibles son muy grandes, y requieren que se den una combinación de factores que son muy poco probables que ocurran en simultáneo para el mismo fixture, por lo que el valor del hipervolumen tenderá a ser muy alto (cerca del 1), incluso en las primeras generaciones. El progreso se produce en general en el segundo o tercer dígito después de la coma.

En la *Figura 5.1* se puede ver cómo el hipervolumen aumenta a medida que se utilizan tamaños de población más grandes. Se puede apreciar también cómo el promedio del hipervolumen de las corridas de tamaño de población de 700 en adelante deja de aumentar con tanta velocidad, lo que permite concluir que tamaños de población aún más grandes no aportarán mejores resultados en contraste de la capacidad de cómputo extra que requerirían. Esto además indica, como era de esperar, que a medida que se obtienen buenas soluciones se hace cada vez más difícil para el algoritmo encontrar soluciones que mejoren los resultados aún más.

Por otro lado, en la *Figura 5.2* se puede apreciar como al aumentar el tamaño de población se estará aumentando el tiempo de ejecución, en una relación lineal. Esto indica que el trabajar con tamaños de población mas grandes, a pesar de brindar mejores resultados, tiene la desventaja de que al algoritmo le tomará más tiempo finalizar.

En el resto del análisis, se utilizó un tamaño de población de 1000, ya que si bien de 700 a 1000 la diferencia en los valores del promedio del hipervolumen es bastante pequeña, de todas formas hay una leve mejoría en lo que a los resultados obtenidos refiere, sobretodo para el caso particular de los breaks. Este objetivo presenta dificultades a la hora de tratar de optimizarlo hasta su valor ideal, que ya habían sido notadas con anterioridad en el análisis del algoritmo de programación entera.

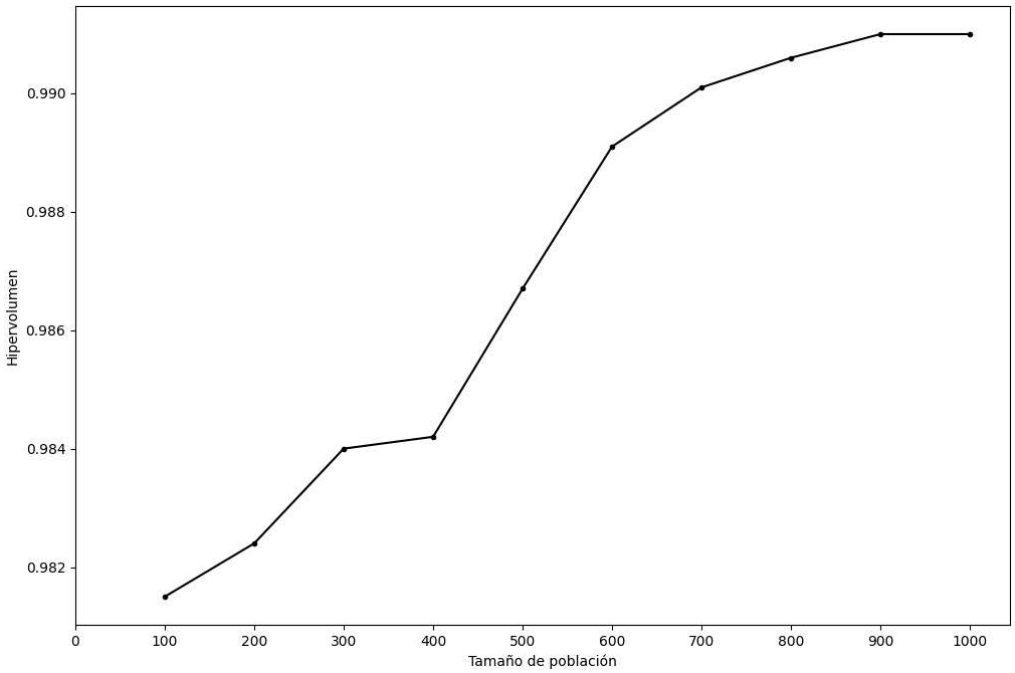

Figura 5.1: Promedio del hipervolumen en 10 ejecuciones para diferentes tamaños de población

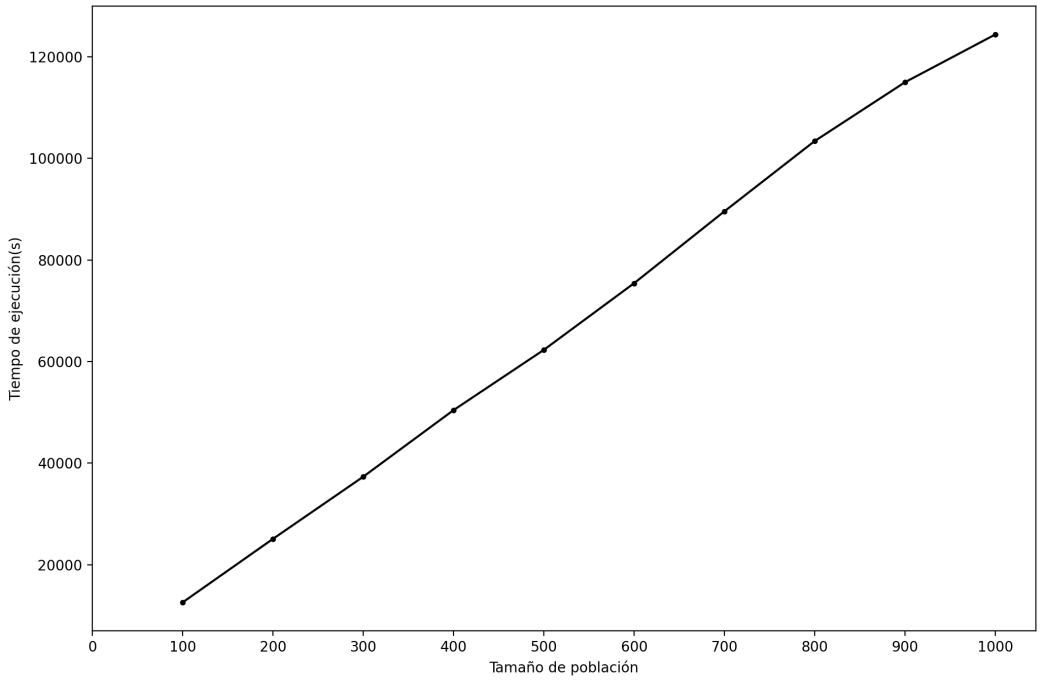

Figura 5.2: Promedio del tiempo en 10 ejecuciones para diferentes tamaños de población

5.3.2. Análisis del efecto de la cantidad de iteraciones

En esta sección se analiza el efecto de la cantidad de iteraciones sobre el desempeño del algoritmo mediante el estudio de la evolución del valor del hipervolumen a lo largo de una ejecución. El algoritmo fue ejecutado con una población de 10000 individuos y se calculó el hipervolumen sobre el frente de Pareto (siguiendo el mismo método de normalización que en la sección anterior) para las generaciones 1000, 2000, 3000, 5000 y 10000.

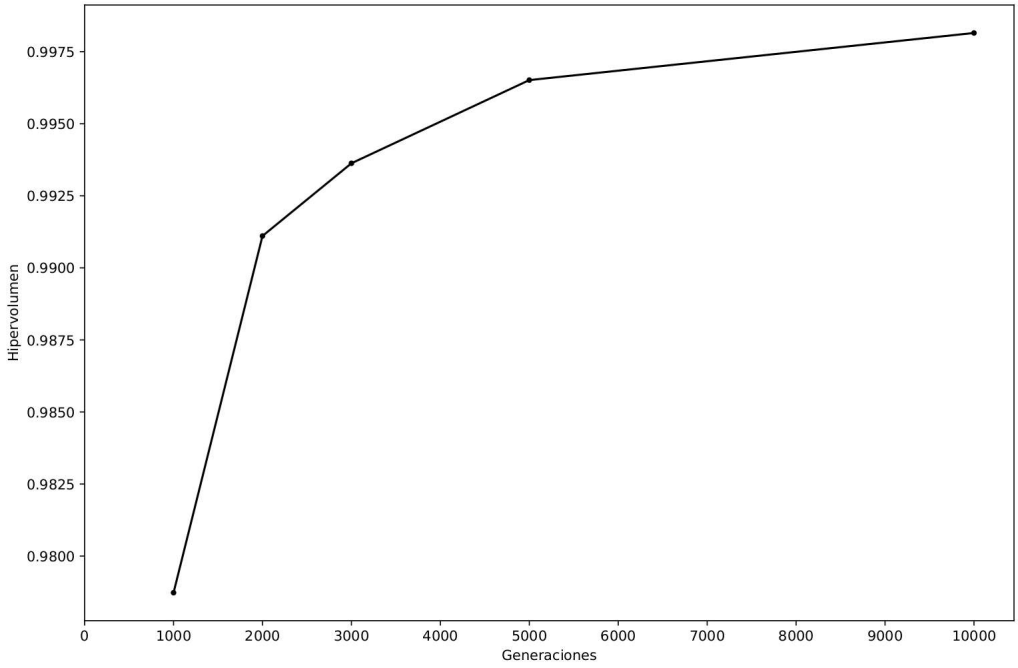

Figura 5.3: Hipervolumen a lo largo de las generaciones

La *Figura 5.3* ilustra la evolución del valor del hipervolumen a lo largo de las diferentes generaciones de la ejecución del algoritmo. Se puede ver que el valor del hipervolumen sube rápidamente en las primeras generaciones (de 1000 a 5000), mientras que en las siguientes generaciones (de 5000 a 10000) se observa un incremento más suave. Esto indica, como se ha podido ver en parte en el análisis de los resultados del algoritmo de programación entera, que a medida que nos acercamos cada vez más al óptimo, es más difícil encontrar soluciones mejores que las ya encontradas por lo que la velocidad de convergencia se ve disminuida. De todas formas, se puede ver que el valor del hipervolumen alcanza un valor muy cercano al punto de referencia, lo que demuestra que las

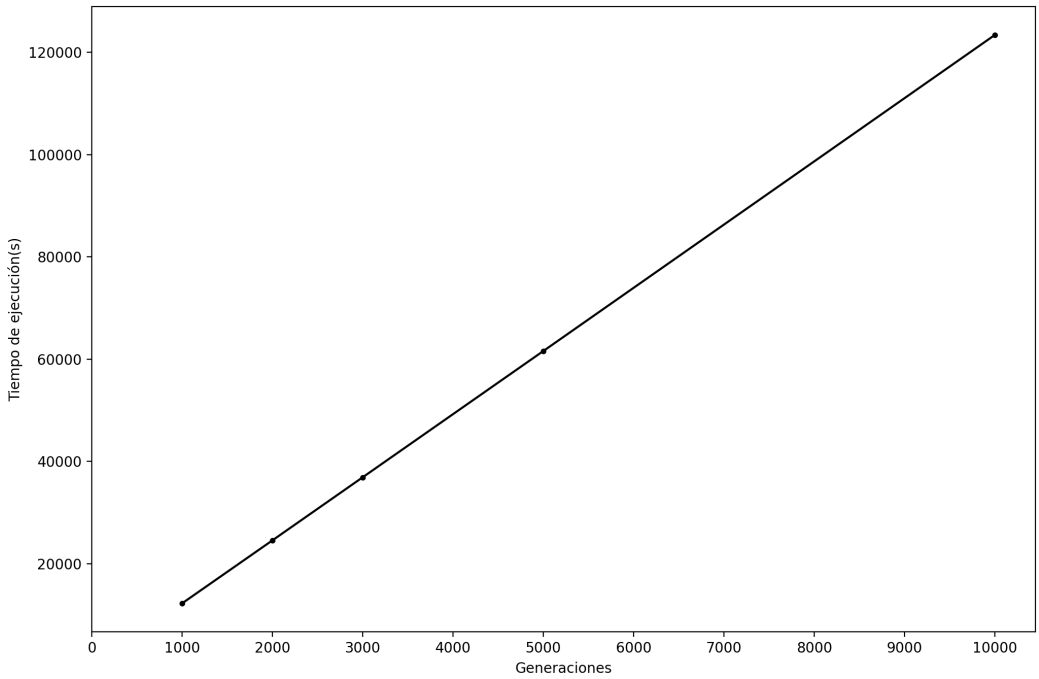

Figura 5.4: Tiempo a lo largo de las generaciones

soluciones obtenidas convergen de buena forma hacia el frente de Pareto.

Por otro lado, al observar la *Figura 5.4* se puede ver el tiempo de ejecución a lo largo de una corrida aumenta de forma lineal, lo cual implica que el agregar más generaciones al algoritmo también incrementará el tiempo de ejecución.

5.3.3. Análisis de convergencia

El objetivo de este análisis es mostrar la convergencia en los distintos objetivos hacia el frente de Pareto a lo largo de la ejecución del algoritmo. Para esto, se presentan gráficas que comparan pares de objetivos, que ilustran el frente de Pareto obtenido en diferentes etapas de la ejecución. El análisis se realiza de a dos objetivos, en lugar de hacerlo con los tres juntos ya que las gráficas resultan más intuitivas y fáciles de comprender.

Una observación importante de aclarar es que no fue incluida la gráfica del par de objetivos *breaks de local - breaks de visitante*, debido a que los valores de ambos objetivos varían y evolucionan de forma similar en el correr de las ejecuciones. Podría llegar a ocurrir en casos específicos alguna diferencia que derive

en que converjan los breaks de local y no los de visitante (o viceversa), pero sería un caso muy particular que no aportaría información relevante al análisis. De todas formas, si se observa la convergencia en todos los pares de objetivos se puede asumir que todos los objetivos están convergiendo simultáneamente.

Se presentan las gráficas de los distintos frentes de Pareto en las *Figuras 5.5* y *5.6* para las generaciones 1000, 2000, 3000, 5000 y 10000 para evitar tener gráficas con demasiados puntos que reduzcan su legibilidad. De todas formas, es una cantidad de veces razonable para poder evaluar su tendencia.

Para estos experimentos se usó una población de 1000 individuos y se dejó evolucionar la población por unas 10000 generaciones. Además, se calcularon los valores óptimos de cada objetivo para el fixture del Campeonato Uruguayo de Fútbol 2021 y se incluyeron en cada gráfica para poder comparar con las soluciones obtenidas.

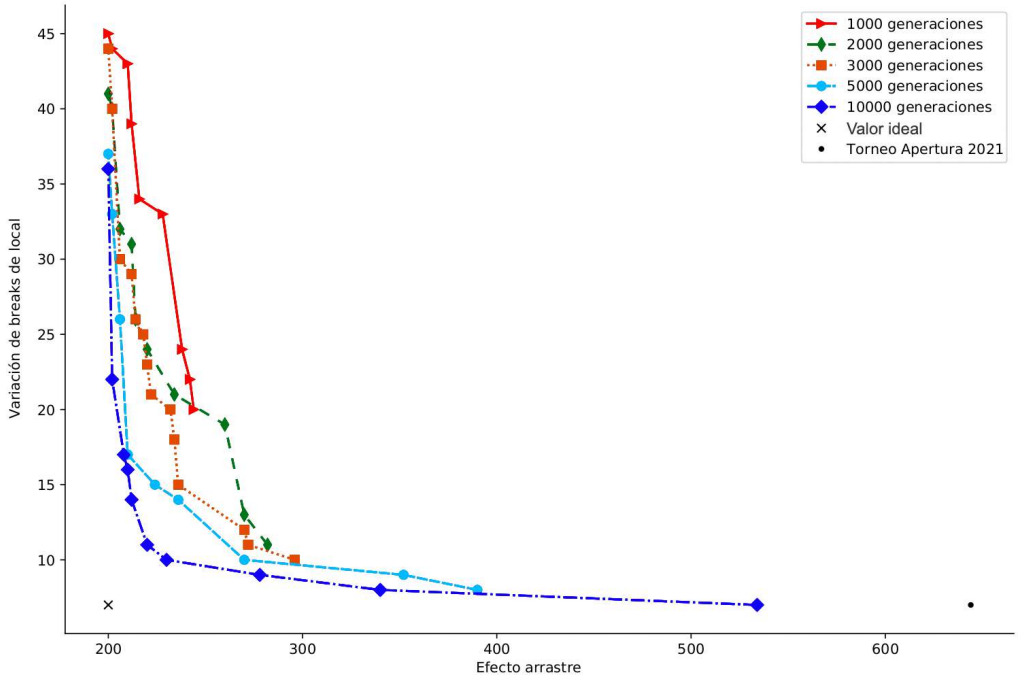

Figura 5.5: Breaks local vs efecto arrastre

Las gráficas de las *Figuras 5.5* y *5.6* dejan ver claramente que el algoritmo es capaz de optimizar todos los objetivos simultáneamente. Se puede ver como

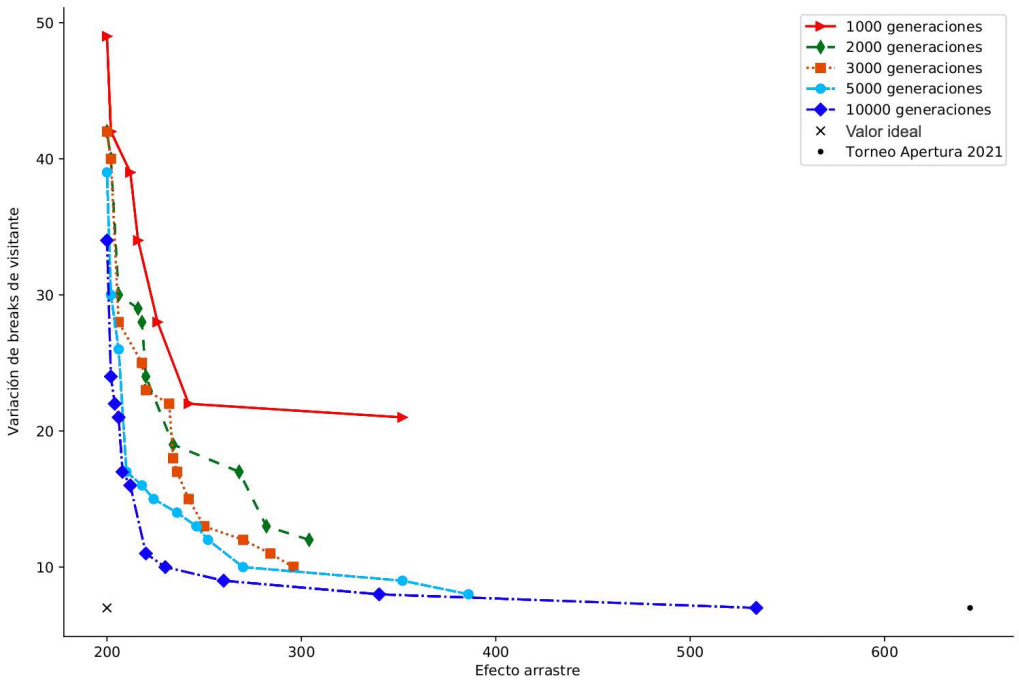

Figura 5.6: Breaks visitante vs efecto arrastre

el frente de Pareto converge hacia el valor ideal, es decir, hacia el punto compuesto por el mínimo valor teórico de cada uno de lo objetivos graficados, a lo largo de las iteraciones, teniendo algunos puntos muy cerca de alcanzar este valor.

Además, se puede observar como el algoritmo obtiene resultados muy variados en cuanto al desempeño en cada uno de los objetivos (por ejemplo, algunas soluciones son mejores en el efecto arrastre pero tienen muchos breaks y viceversa). A medida que se llega al numero máximo de iteraciones, se puede ver que el frente de Pareto es cada vez más amplio, lo que indica que se encontraron soluciones que son extremadamente buenas en los distintos objetivos (algunas incluso teniendo el valor óptimo en uno de ellos) pero a costa de un mal desempeño en el otro. Esto muestra claramente que los objetivos considerados están en conflicto, cuando un objetivo mejora, en muchas ocasiones el otro objetivo empeora. Sin embargo se pueden ver soluciones muy buenas en ambos objetivos (muy cercanas a los valores óptimos).

Para cada ejecución existe un subconjunto de soluciones que logran balan-

cear ambos objetivos simultáneamente, logrando valores de breaks de local y visitante cercanos a 1 por equipo, resultando en un valor total de breaks bajo, a la vez que obteniendo un valor de efecto arrastre sumamente bajo.

5.3.4. Análisis de los resultados obtenidos

En esta sección, el objetivo es analizar los resultados del algoritmo genético comparándolo contra los fixtures reales utilizados y los obtenidos por el algoritmo de programación entera. Para esto se presentarán varias tablas, las cuales muestran la cantidad de breaks y el efecto arrastre encontrados en los fixtures reales, los obtenidos por el enfoque de programación entera al minimizar breaks y al minimizar efecto arrastre, y el algoritmo NSGA-II (la mejor solución con respecto a los breaks y al efecto arrastre, así como un último fixture que balancea ambos objetivos a la vez). Para el caso del fixture del Campeonato Uruguayo de Fútbol, se comparan los fixtures obtenidos con los del Campeonato Uruguayo de Fútbol 2020 y 2021 en la *Tabla 5.4*. En cuanto a la Liga Uruguaya de Básquetbol, se muestran los fixtures obtenidos en dos tablas distintas (*Tablas 5.5 y 5.6*) ya que los fixtures del año 2020/21 y 2021/22 contaban con una cantidad distinta de equipos (doce y catorce, respectivamente). Los fixtures reales y los obtenidos con los algoritmos se encuentran en detalle en las secciones A y B del apéndice, respectivamente.

En la *Tabla 5.4* se puede ver que el algoritmo NSGA-II fue capaz de encontrar una solución (fixture presentado en la *Tabla A2.7*) que iguala en cantidad de breaks a los fixtures del Campeonato Uruguayo de Fútbol 2020 y 2021 (fixtures presentados en las *Tablas A1.1* y *A1.2*, respectivamente) pero mejora en efecto arrastre. Nótese que estos tres fixtures tienen el mínimo de breaks posibles para un fixture de 16 equipos. Por otro lado, comparando contra la mejor solución en breaks del algoritmo de programación entera (fixture presentado en la *Tabla A2.1*), se puede ver que la mejor solución en breaks generada por algoritmo genético es superior en cuanto a breaks, mientras que es bastante inferior en cuanto a efecto arrastre.

Al observar los fixtures en detalle incluidos en el apéndice, se puede apreciar cómo no hay prácticamente breaks en estos cuatro fixtures mencionados e incluso algunos equipos no presentan breaks siquiera, como ser el caso de Dep.

Maldonado y Boston River en el Campeonato Uruguayo de Fútbol 2021, de Liverpool y Boston River en el Campeonato Uruguayo de Fútbol 2020 y de Liverpool y Mvd. City Torque en el fixture generado por el algoritmo NSGA-II. Sin embargo, el hecho de que tengan altos valores de efecto de arrastre se traduce en equipos que se benefician en alta medida del fixture de partidos. Por ejemplo, como se menciona anteriormente en la sección 5.2, Danubio y Nacional se ven beneficiados por el efecto de arrastre en trece fechas en el Campeonato Uruguayo de Fútbol 2020, mientras que en el caso del Campeonato Uruguayo de Fútbol 2021 se tiene el caso de Dep. Maldonado que se beneficia de dicho efecto en trece fechas. A su vez, en el caso del fixture generado por el algoritmo NSGA-II, Peñarol se ve beneficiado en trece fechas. Esto contrasta con la gran mayoría de los equipos que no se beneficia del efecto de arrastre en ninguna fecha, lo cual vuelve a estos fixtures bastante desbalanceados.

En cuanto a la mejor solución obtenida por el algoritmo NSGA-II con respecto al efecto arrastre (fixture presentado en la *Tabla A2.8*), se puede ver que es ampliamente superior en cuanto a ese valor que los fixtures reales del Campeonato Uruguayo. Además, al compararse con la solución que minimiza el efecto arrastre del algoritmo de programación entera (fixture presentado en la *Tabla A2.2*), demuestra ser mejor tanto en efecto arrastre como en breaks (200 vs 202 en cuanto a efecto arrastre y 44 vs 102 en cuanto a los breaks). Si se observan los fixtures en detalle incluidos en el apéndice se puede ver como, tanto la mejor solución obtenida por el algoritmo NSGA-II con respecto al efecto arrastre como la solución que minimiza este mismo valor obtenida con el enfoque de programación entera, presentan valores muy bajos de efecto arrastre, Más específicamente, se puede ver como la gran mayoría de los equipos es beneficiado en tres o cuatro ocasiones por dicho efecto. Sin embargo, sí se pueden notar altos valores en cuanto a breaks. En el caso del fixture generado por el enfoque de programación entera, algunos equipos llegan a presentar altas cantidades de breaks como Mvd. City Torque que tiene ocho (incluyendo una serie de cinco partidos seguidos de visitante en las últimas cinco fechas). En el caso del fixture generado por el algoritmo NSGA-II no se observan casos de equipos con breaks muy altos si no que la gran mayoría presenta dos o tres breaks.

Por último, se considera una solución del algoritmo genético que balancea

relativamente bien ambos objetivos (fixture presentado en la *Tabla A2.9*), elegida por tener la mínima distancia con respecto al punto ideal en comparación con el resto de los individuos del frente de Pareto final. Esta solución sacrifica mínimamente el desempeño en breaks por un mejor desempeño en cuanto al efecto arrastre. En este caso, al observar el fixture en detalle se puede ver como es bastante balanceado tanto en breaks como en efecto arrastre, teniendo un promedio de dos y cuatro beneficios de efecto arrastre por equipo. Se observan sin embargo algunos equipos que no tienen breaks (Mvd. City Torque y Fénix) mientras algunos presentan tres (como Cerrito y Cerro Largo), por lo que el fixture aún tiene espacio a mejorar para ser más balanceado.

Tabla 5.4: Comparación de fixtures para el Campeonato Uruguayo de Fútbol

Fixtures reales	**Breaks**	**efecto arrastre**
Campeonato Uruguayo de Fútbol 2020	**14**	620
Campeonato Uruguayo de Fútbol 2021	**14**	644
Fixtures obtenidos con algoritmo de PE		
Minimizando breaks	16	252
Minimizando efecto arrastre	102	202
Fixtures obtenidos con algoritmo evolutivo		
Mejor en breaks	**14**	534
Mejor en efecto arrastre	44	**200**
Balanceado	24	212

Valores óptimos marcados en negrita

Tabla 5.5: Comparación de fixtures para la Liga Uruguaya de Básquetbol del año 2020/21

Fixtures reales	**Breaks**	**efecto arrastre**
Liga Uruguaya de Básquetbol 2020/21	34	152
Fixtures obtenidos con algoritmo de PE		
Minimizando breaks	**10**	158
Minimizando efecto arrastre	56	**136**
Fixtures obtenidos con algoritmo evolutivo		
Mejor en breaks	**10**	154
Mejor en efecto arrastre	18	**136**
Balanceado	12	140

Valores óptimos marcados en negrita

La *Tabla 5.5* permite observar que, para la Liga Uruguaya de Básquetbol 2020/21, tanto la solución obtenida por el algoritmo de programación entera que minimiza los breaks (fixture presentado en la *Tabla A2.3*), como la solución que mejor desempeño muestra en cuanto a breaks de las obtenidas con el algoritmo NSGA-II (fixture presentado en la *Tabla A2.10*), son bastante superiores en términos de breaks (ambas alcanzan el valor óptimo) que el fixture real (fixture presentado en la *Tabla A1.3*). De hecho, ambas soluciones consiguen minimizar la cantidad de breaks. Más aún, se puede ver que la solución obtenida con el algoritmo genético es incluso preferible ante la obtenida con el algoritmo de programación entera ya que para una misma cantidad de breaks logra un menor valor del efecto arrastre. Al analizar los fixtures en detalle, se puede ver que el fixture de la Liga Uruguaya de Básquetbol 2020/21 presenta una alta cantidad de breaks, llegando a tener equipos con 5 breaks como es el caso de Nacional. En cambio, las soluciones que minimizan los breaks obtenidas con los algoritmos estudiados en este trabajo presentan la cantidad mínima de breaks, por lo que todos los equipos tienen equipos un solo break (a excepción de dos que no presentan breaks: Nacional y Trouville en el caso del fixture obtenido con programación entera y Trouville y Goes en el caso del fixture obtenido con el algoritmo NSGA-II).

Con respecto al efecto arrastre, las soluciones que mejor desempeño lograron para este valor en cada algoritmo (*Tabla A2.4* para el fixture obtenido con el enfoque de programación entera y *A2.11* para el fixture obtenido con el algoritmo NSGA-II) mejoran al fixture real de la Liga Uruguaya de Básquetbol con respecto a este objetivo, llegando incluso a obtener el mínimo valor posible. Adicionalmente, la solución obtenida con el algoritmo genético que mejor resultado obtuvo en términos de efecto arrastre no sólo mejora al fixture real en efecto arrastre, si no que incluso demuestra un mejor desempeño en cuanto a la cantidad de breaks. En los fixtures detallados en el apéndice, se puede apreciar que en estas soluciones la gran mayoría de los equipos se benefician tres veces del efecto arrastre, mientras que algunos pocos lo hacen unas cuatro veces. Sin embargo, también se puede observar como, a diferencia de la solución obtenida con el algoritmo NSGA-II, la obtenida con el algoritmo de programación entera presenta equipos con muchos breaks, como por ejemplo Def. Sporting con siete breaks (incluyendo una serie de partidos de visitante en las últimas cinco fechas) u Olimpia con siete breaks (incluyendo una corrida

de partidos de visitante seguidos de la fecha tres a la nueve).

El algoritmo NSGA-II es capaz de encontrar una solución que logra un muy buen balance entre la cantidad de breaks y el efecto arrastre (fixture presentado en la *Tabla A2.12*), ya que solamente sacrifica un par de breaks para obtener un valor muy bajo en efecto arrastre.

Tabla 5.6: Comparación de fixtures para la Liga Uruguaya de Básquetbol del año 2021/22

Fixtures reales	**Breaks**	**Efecto arrastre**
Liga Uruguaya de Básquetbol 2021/22	54	224
Fixtures obtenidos con algoritmo de PE		
Minimizando breaks	**12**	238
Minimizando efecto arrastre	76	**168**
Fixtures obtenidos con algoritmo evolutivo		
Mejor en breaks	**12**	184
Mejor en efecto arrastre	32	**168**
Balanceado	18	176

Valores óptimos marcados en negrita

La *Tabla 5.6* deja ver que, tanto la solución que minimiza los breaks del algoritmo de programación entera (fixture presentado en la *Tabla A2.5*), como la solución que mejor desempeño en breaks obtuvo del algoritmo NSGA-II (fixture presentado en la *Tabla A2.13*), alcanzan los valores óptimos de breaks y por consiguiente tienen un mejor valor de breaks que el fixture real de la Liga Uruguaya de Básquetbol 2021/22 (fixture presentado en la *Tabla A1.4*). Adicionalmente, la solución obtenida con el algoritmo NSGA-II obtuvo también un valor de efecto arrastre inferior al fixture real. Analizando los fixtures reales en detalle, se aprecia como el fixture de la Liga Uruguaya de Básquetbol 2021/22 presenta una alta cantidad de breaks, teniendo muchos de sus equipos con cinco breaks (como Olimpia o Goes, por ejemplo). Además, también se pueden observar algunos equipos que se benefician en alta medida del efecto arrastre, como es el caso de Olivol Mundial que lo hace en ocho fechas. Mientras tanto, los fixtures obtenidos con los algoritmos estudiados en este trabajo, al tener la cantidad mínima de breaks posible, logran que todos los equipos tengan un solo break a excepción de dos que no presentan ninguno (Peñarol y Urunday en el caso del fixture obtenido con programación entera, Hebraica

Macabi y Biguá en el obtenido con el algoritmo NSGA-II). Sin embargo, en el fixture obtenido con el algoritmo de programación entera también se pueden observar varios equipos que se benefician bastante del efecto arrastre, como es el caso de Hebraica Macabi que se ve beneficiado por dicho efecto en nueve fechas. Este no es el caso del fixture obtenido por el algoritmo NSGA-II, que logra balancear un poco mejor las ocasiones en las que los equipos se benefician de este efecto, en promedio tres o cuatro veces por equipo.

Por otro lado, en cuanto al efecto arrastre, la solución obtenida con el algoritmo de programación entera que minimiza este valor (fixture presentado en la *Tabla A2.6*) y la solución obtenida con el algoritmo NSGA-II que mejor desempeño alcanzó en dicho valor (fixture presentado en la *Tabla A2.14*) también demuestran ser superiores que el fixture real de la Liga Uruguaya de Básquetbol 2021/22 (ambas soluciones alcanzan el valor óptimo). Más aún, la solución con mejor valor en efecto arrastre obtenida con el algoritmo NSGA-II es incluso mejor en términos de breaks que las otras dos. Se puede observar también como ambas soluciones obtenidas con el algoritmo NSGA-II son mejores en uno de los objetivos pero peor en el otro que su contraparte, lo que indica, como se vio previamente, que el mejorar en términos de un objetivo afecta el rendimiento sobre el otro. Analizando estos fixtures obtenidos, se observa como el obtenido por el algoritmo de programación entera, a pesar de tener el valor mínimo en efecto arrastre, tiene equipos con una cantidad alta de breaks, como por ejemplo Nacional con ocho breaks. En el caso del fixture obtenido con el algoritmo NSGA-II, no se observan equipos con altas cantidades de breaks, si no que por el contrario, la mayoría de los equipos presentan dos o tres breaks.

En esta tabla también se lista una solución que balancea el desempeño en ambos objetivos (fixture presentado en la *Tabla A2.15*), sacrificando ligeramente el desempeño en cuanto a breaks para mejorar un poco en cuanto a efecto arrastre.

Capítulo 6

Conclusiones y trabajo futuro

El objetivo principal de este proyecto fue la elaboración de un algoritmo que permita la creación de un fixture para los campeonatos de fútbol en nuestro país, de manera de mejorar a los utilizados al día de hoy.

En primera instancia, se relevaron una serie de artículos que fueron de gran importancia para conocer el estado del arte actual en lo que respecta a elaboración de fixtures deportivos por medio de técnicas de optimización. Luego de dicho relevamiento, se trabajó en transformar la realidad del fútbol uruguayo con todos sus requerimientos y objetivos, a un modelado matemático que fue utilizado de base en las etapas siguientes.

Como primera aproximación al problema, se implementó un algoritmo de programación entera, que utilizando el solver matemático Gurobi, permitió optimizar por separado, tanto los breaks como el efecto arrastre. Para ambos objetivos se obtuvieron buenos resultados, alcanzando los valores óptimos para los torneos de básquetbol (torneos de doce y catorce equipos), y quedando muy cerca del valor óptimo en el caso del fútbol (torneo de dieciséis equipos). Un punto importante a destacar es que en el caso del fútbol, a pesar de no haber llegado al valor óptimo al minimizar los breaks, se pudo disminuir considerablemente el valor del efecto arrastre con respecto a los fixtures reales considerados en el análisis.

Luego de diseñado el algoritmo de programación entera recién descrito, se siguió con el enfoque de considerar ambos objetivos en simultáneo. Para ello,

y basados en el relevamiento de estado del arte ya mencionado, se optó por incorporar el uso de metaheurísticas para el diseño del nuevo algoritmo multi-objetivo, en particular se escogió diseñar un algoritmo genético de tipo NSGA-II. Este algoritmo, al igual que el de programación entera, generó muy buenos resultados, tanto para el fútbol como para el básquetbol, llegando en todos los casos a los valores óptimos, incluso para el caso de los breaks en el fixture del fútbol. Incluso para los mejores fixtures del básquetbol en cuanto al efecto arrastre, también se mejoró la cantidad de breaks en comparación con los fixtures reales, por lo que se puede concluir que el algoritmo multi-objetivo diseñado es altamente exitoso.

Tal como se presentó en el capítulo anterior, los resultados obtenidos fueron muy positivos, generando soluciones factibles con buenos valores para los dos objetivos principales considerados. Además, teniendo en cuenta estos dos objetivos en simultáneo, se puede concluir que se mejoraron los últimos fixtures reales, sobre todo en lo que respecta al efecto arrastre. Este aspecto indudablemente no ha sido considerado por la AUF en la realización de los fixtures para el campeonato uruguayo, a pesar de ayudar en buena medida a lograr un mayor grado de justicia posible. En el caso de los fixtures de la LUB, ocurre algo similar con respecto a los breaks, presentando valores considerablemente altos con respecto a las soluciones obtenidas. Estos hechos muestran que el uso de técnicas de optimización en este contexto, es altamente aplicable para la elaboración de fixtures de campeonatos deportivos en Uruguay.

A pesar de haber conseguido resultados que pueden ser considerados ampliamente exitosos en comparación con los fixtures reales que se han utilizado en los últimos años, es posible señalar una serie de aspectos que es posible mejorar y complementar en este proyecto.

Como se explicó anteriormente, a la hora de desarrollar un fixture, sólo se tuvo en cuenta los torneos Apertura y Clausura, por lo que un punto que puede ser tenido en cuenta en el futuro, es considerar también el torneo Intermedio. Esto llevará también a modificar el modelado del problema actual, añadiendo nuevas consideraciones específicas de este nuevo torneo disputado entre el torneo Apertura y el Clausura, así como también modificando restricciones y objetivos actuales que pueden llegar a perder sentido con los nuevos encuen-

tros a ser disputados. Por otra parte, en los próximos años es posible que se comience a disputar un nuevo campeonato, denominado Copa Uruguay, participando equipos de distintas divisiones, profesionales y amateurs, incluyendo también una gran cantidad de equipos del interior del país. Dado este nuevo contexto, objetivos como la minimización de las distancias viajadas por cada equipo pueden volver a tener una importancia preponderante en la elaboración de los fixtures, con el objetivo de no perjudicar, sobre todo a los equipos del interior.

En el análisis de la solución propuesta, la totalidad de las pruebas realizadas utilizaron como base el modelo uno, es decir únicamente considerando un torneo de una vuelta (torneo Apertura). Sin embargo, con el objetivo de conseguir un fixture más completo y personalizable, también es interesante ofrecer la posibilidad de considerar el torneo Clausura a la hora de crear el fixture. Esto es posible realizarlo mediante el uso del modelo dos, que además de crear un fixture doble round robin, también permite tener un orden de fechas distinto para el torneo Apertura y Clausura, sin seguir ningún tipo de esquema espejo o similar. Finalmente, se podría extender aún más el diseño del algoritmo incorporando el modelo tres, considerando un fixture doble round robin con los partidos asignados a un día específico de cada fecha.

Por otra parte, puede ser de interés interactuar con la Asociación Uruguaya de Fútbol, con el objetivo de lograr un intercambio de información. Por un lado, para complementar el modelado antes presentado, comprendiendo de mejor manera la realidad del fútbol uruguayo desde la perspectiva de la AUF y los equipos. Esto permitirá conocer nuevos objetivos y restricciones que pueden no haber sido tenidos en cuenta en este trabajo, permitiendo ajustar el enfoque seguido en el proyecto. Por otro lado, también sería interesante poder compartir con la AUF el conocimiento y las soluciones conseguidas a lo largo de este trabajo, con el objetivo de mostrar las ventajas del enfoque seguido en este trabajo con respecto a la metodología actual de elaboración de fixtures.

Por otro lado, dado el relevamiento de artículos realizado, se consideró apropiado utilizar un Algoritmo Genético para la implementación de un algoritmo multi-objetivo que permita optimizar simultáneamente todos los objetivos considerados. Sin embargo, la utilización de otras técnicas metaheurísticas permi-

tiría ampliar la gama de posibilidades existentes, pudiéndose llegar a generar soluciones mejores que las alcanzadas en este trabajo.

Un punto en contra a la hora de resolver este tipo de problemas, es que debido a su alta complejidad computacional, los tiempos de ejecución pueden llegar a ser muy altos. Por ello, tanto en el algoritmo implementado con Gurobi, como en el algoritmo genético, se pueden estudiar posibles mejoras y ajustes en los distintos parámetros de los algoritmos utilizados para lograr una mejora en la performance en cada ejecución.

Referencias bibliográficas

Alarcón, F., Durán, G., Guajardo, M., Miranda, J., Muñoz, H., Ramírez, L., Ramírez, M., Sauré, D., Siebert, M., Souyris, S., et al. (2017). Operations research transforms the scheduling of chilean soccer leagues and south american world cup qualifiers. *Interfaces*, 47(1):52–69.

Barone, L., While, L., Hughes, P., and Hingston, P. (2006). Fixture-scheduling for the australian football league using a multi-objective evolutionary algorithm. In *2006 IEEE International Conference on Evolutionary Computation*, pages 954–961. IEEE.

Bonomo, F., Cardemil, A., Durán, G., Marenco, J., and Sabán, D. (2012). An application of the traveling tournament problem: The argentine volleyball league. *Interfaces*, 42(3):245–259.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.

Denysiuk, R. (2014). *Evolutionary Multiobjective Optimization: Review, Algorithms, and Applications*. PhD thesis.

Dinitz, J. H., Froncek, D., Lamken, E. R., and Wallis, W. D. (2006). Scheduling a tournament. In *Handbook of Combinatorial Designs*, pages 617–631. Chapman and Hall/CRC.

Durán, G., Durán, S., Marenco, J., Mascialino, F., and Rey, P. A. (2019). Scheduling argentina's professional basketball leagues: A variation on the travelling tournament problem. *European Journal of Operational Research*, 275(3):1126–1138.

Durán, G., Guajardo, M., and Sauré, D. (2017). Scheduling the south american qualifiers to the 2018 fifa world cup by integer programming. *European Journal of Operational Research*, 262(3):1109–1115.

Easton, K., Nemhauser, G., and Trick, M. (2001). The traveling tournament problem description and benchmarks. In *International Conference on Principles and Practice of Constraint Programming*, pages 580–584. Springer.

Goldberg, D. E. (2006). *Genetic algorithms*. Pearson Education India.

Goossens, D. and Spieksma, F. (2009). Scheduling the belgian soccer league. *Interfaces*, 39(2):109–118.

Gurobi Optimization, LLC (2022). Gurobi Optimizer Reference Manual. `https://www.gurobi.com`.

Kyngäs, J., Nurmi, K., Kyngäs, N., Lilley, G., Salter, T., and Goossens, D. (2017). Scheduling the australian football league. *Journal of the Operational Research Society*, 68(8):973–982.

Recalde, D., Torres, R., and Vaca, P. (2013). Scheduling the professional ecuadorian football league by integer programming. *Computers & Operations Research*, 40(10):2478–2484.

Régin, J.-C. (2001). Minimization of the number of breaks in sports scheduling problems using constraint programming. *DIMACS series in discrete mathematics and theoretical computer science*, 57:115–130.

Thielen, C. and Westphal, S. (2011). Complexity of the traveling tournament problem. *Theoretical Computer Science*, 412(4-5):345–351.

van't Hof, P., Post, G., and Briskorn, D. (2009). Round-robin tournaments with minimum number of breaks and two teams per club. Technical report, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.

While, L. and Kendall, G. (2014). Scheduling the english football league with a multi-objective evolutionary algorithm. In *International Conference on Parallel Problem Solving from Nature*, pages 842–851. Springer.

APÉNDICES

Apéndice 1

Fixtures reales

1.1. Campeonato Uruguayo de Fútbol 2020

Tabla A1.1: Fixture del Campeonato Uruguayo de Fútbol 2020

FECHA 1	**FECHA 2**
Mvd. City Torque - Progreso	Dep. Maldonado - Liverpool
River Plate - Fénix	Boston River - Rentistas
Peñarol - Cerro	Nacional - Cerro Largo
Wanderers - Def. Sporting	Danubio - Wanderers
Cerro Largo - Danubio	Def. Sporting - Peñarol
Rentistas - Nacional	Cerro - River Plate
Dep. Maldonado - Boston River	Fénix - Mvd. City Torque
Liverpool - Plaza Colonia	Progreso - Plaza Colonia
FECHA 3	**FECHA 4**
Liverpool - Progreso	Rentistas - Liverpool
Plaza Colonia - Fénix	Dep. Maldonado - Cerro Largo
Mvd. City Torque - Cerro	Boston River - Wanderers
River Plate - Def. Sporting	Nacional - Peñarol
Peñarol - Danubio	Danubio - River Plate
Wanderers - Nacional	Def. Sporting - Mvd. City Torque
Cerro Largo - Boston River	Cerro - Plaza Colonia
Rentistas - Dep. Maldonado	Fénix - Progreso
FECHA 5	**FECHA 6**
Liverpool - Fénix	Cerro Largo - Liverpool

Progreso - Cerro	Rentistas - Wanderers
Plaza Colonia - Def. Sporting	Dep. Maldonado - Peñarol
Mvd. City Torque - Danubio	Boston River - River Plate
River Plate - Nacional	Nacional - Mvd. City Torque
Peñarol - Boston River	Danubio - Plaza Colonia
Wanderers - Dep. Maldonado	Def. Sporting - Progreso
Cerro Largo - Rentistas	Cerro - Fénix
FECHA 7	**FECHA 8**
Liverpool - Cerro	Wanderers - Liverpool
Fénix - Def. Sporting	Cerro Largo - Peñarol
Progreso - Danubio	Rentistas - River Plate
Plaza Colonia - Nacional	Dep. Maldonado - Mvd. City Torque
Mvd. City Torque - Boston River	Boston River - Plaza Colonia
River Plate - Dep. Maldonado	Nacional - Progreso
Peñarol - Rentistas	Danubio - Fénix
Wanderers - Cerro Largo	Def. Sporting - Cerro
FECHA 9	**FECHA 10**
Liverpool - Def. Sporting	Peñarol - Liverpool
Cerro - Danubio	Wanderers - River Plate
Fénix - Nacional	Cerro Largo - Mvd. City Torque
Progreso - Boston River	Rentistas - Plaza Colonia
Plaza Colonia - Dep. Maldonado	Dep. Maldonado - Progreso
Mvd. City Torque - Rentistas	Boston River - Fénix
River Plate - Cerro Largo	Nacional - Cerro
Peñarol - Wanderers	Danubio - Def. Sporting
FECHA 11	**FECHA 12**
Liverpool - Danubio	River Plate - Liverpool
Def. Sporting - Nacional	Peñarol - Mvd. City Torque
Cerro - Boston River	Wanderers - Plaza Colonia
Fénix - Dep. Maldonado	Cerro Largo - Progreso
Progreso - Rentistas	Rentistas - Fénix
Plaza Colonia - Cerro Largo	Dep. Maldonado - Cerro
Mvd. City Torque - Wanderers	Boston River - Def. Sporting
River Plate - Peñarol	Nacional - Danubio
FECHA 13	**FECHA 14**
Liverpool - Nacional	Mvd. City Torque - Liverpool

Danubio - Boston River	River Plate - Plaza Colonia
Def. Sporting - Dep. Maldonado	Peñarol - Progreso
Cerro - Rentistas	Wanderers - Fénix
Fénix - Cerro Largo	Cerro Largo - Cerro
Progreso - Wanderers	Rentistas - Def. Sporting
Plaza Colonia - Peñarol	Dep. Maldonado - Danubio
Mvd. City Torque - River Plate	Boston River - Nacional

FECHA 15
Liverpool - Boston River
Nacional - Dep. Maldonado
Danubio - Rentistas
Def. Sporting - Cerro Largo
Cerro - Wanderers
Fénix - Peñarol
Progreso - River Plate
Plaza Colonia - Mvd. City Torque

1.2. Campeonato Uruguayo de Fútbol 2021

Tabla A1.2: Fixture del Campeonato Uruguayo de Fútbol 2021

FECHA 1	**FECHA 2**
Dep. Maldonado - Liverpool	Wanderers - Dep. Maldonado
Cerrito - River Plate	Boston River - Rentistas
Nacional - Cerro Largo	Fénix - Peñarol
Villa Española - Progreso	Plaza Colonia - Mvd. City Torque
Mvd. City Torque - Sud América	Sud América - Villa Española
Peñarol - Plaza Colonia	Progreso - Nacional
Rentistas - Fénix	Cerro Largo - Cerrito
Wanderers - Boston River	River Plate - Liverpool
FECHA 3	**FECHA 4**
Dep. Maldonado - River Plate	Rentistas - Dep. Maldonado
Liverpool - Cerro Largo	Wanderers - Peñarol
Cerrito - Progreso	Boston River - Mvd. City Torque
Nacional - Sud América	Fénix - Villa Española
Villa Española - Plaza Colonia	Plaza Colonia - Nacional
Mvd. City Torque - Fénix	Sud América - Cerrito
Peñarol - Boston River	Progreso - Liverpool
Rentistas - Wanderers	Cerro Largo - River Plate
FECHA 5	**FECHA 6**
Dep. Maldonado - Cerro Largo	Peñarol - Dep. Maldonado
River Plate - Progreso	Rentistas - Mvd. City Torque
Liverpool - Sud América	Wanderers - Villa Española
Cerrito - Plaza Colonia	Boston River - Nacional
Nacional - Fénix	Fénix - Cerrito
Villa Española - Boston River	Plaza Colonia - Liverpool
Mvd. City Torque - Wanderers	Sud América - River Plate
Peñarol - Rentistas	Progreso - Cerro Largo
FECHA 7	**FECHA 8**
Dep. Maldonado - Progreso	Mvd. City Torque - Dep. Maldonado
Cerro Largo - Sud América	Peñarol - Villa Española
River Plate - Plaza Colonia	Rentistas - Nacional
Liverpool - Fénix	Wanderers - Cerrito

Cerrito - Boston River	Boston River - Liverpool
Nacional - Wanderers	Fénix - River Plate
Villa Española - Rentistas	Plaza Colonia - Cerro Largo
Mvd. City Torque - Peñarol	Sud América - Progreso
FECHA 9	**FECHA 10**
Dep. Maldonado - Sud América	Villa Española - Dep. Maldonado
Progreso - Plaza Colonia	Mvd. City Torque - Nacional
Cerro Largo - Fénix	Peñarol - Cerrito
River Plate - Boston River	Rentistas - Liverpool
Liverpool - Wanderers	Wanderers - River Plate
Cerrito - Rentistas	Boston River - Cerro Largo
Nacional - Peñarol	Fénix - Progreso
Villa Española - Mvd. City Torque	Plaza Colonia - Sud América
FECHA 11	**FECHA 12**
Dep. Maldonado - Plaza Colonia	Nacional - Dep. Maldonado
Sud América - Fénix	Villa Española - Cerrito
Progreso - Boston River	Mvd. City Torque - Liverpool
Cerro Largo - Wanderers	Peñarol - River Plate
River Plate - Rentistas	Rentistas - Cerro Largo
Liverpool - Peñarol	Wanderers - Progreso
Cerrito - Mvd. City Torque	Boston River - Sud América
Nacional - Villa Española	Fénix - Plaza Colonia
FECHA 13	**FECHA 14**
Dep. Maldonado - Fénix	Cerrito - Dep. Maldonado
Plaza Colonia - Boston River	Nacional - Liverpool
Sud América - Wanderers	Villa Española - River Plate
Progreso - Rentistas	Mvd. City Torque - Cerro Largo
Cerro Largo - Peñarol	Peñarol - Progreso
River Plate - Mvd. City Torque	Rentistas - Sud América
Liverpool - Villa Española	Wanderers - Plaza Colonia
Cerrito - Nacional	Boston River - Fénix

FECHA 15
Dep. Maldonado - Boston River
Fénix - Wanderers
Plaza Colonia - Rentistas

Sud América - Peñarol
Progreso - Mvd. City Torque
Cerro Largo - Villa Española
River Plate - Nacional
Liverpool - Cerrito

1.3. Liga Uruguaya de Básquetbol 2020/21

Tabla A1.3: Fixture de la Liga Uruguaya de Básquetbol 2020/21

FECHA 1	**FECHA 2**
Trouville - Malvín	Trouville - Capitol
Hebraica Macabi - Goes	Def. Sporting - Hebraica Macabi
Capitol - Biguá	Goes - Olimpia
Olimpia - Def. Sporting	Malvín - Aguada
Nacional - Peñarol	Peñarol - Urunday U.
Urunday U. - Aguada	Biguá - Nacional
FECHA 3	**FECHA 4**
Hebraica Macabi - Urunday U.	Nacional - Olimpia
Malvín - Goes	Capitol - Malvín
Def. Sporting - Nacional	Trouville - Peñarol
Aguada - Trouville	Urunday U. - Def. Sporting
Peñarol - Capitol	Aguada - Hebraica Macabi
Olimpia - Biguá	Goes - Biguá
FECHA 5	**FECHA 6**
Olimpia- Urunday U.	Hebraica Macabi - Capitol
Malvín - Hebraica Macabi	Def. Sporting - Malvín
Nacional - Trouville	Trouville - Olimpia
Capitol - Goes	Goes - Peñarol
Peñarol - Aguada	Aguada - Nacional
Biguá - Def. Sporting	Biguá - Urunday U.
FECHA 7	**FECHA 8**
Olimpia - Peñarol	Malvín - Urunday U.
Urunday U. - Trouville	Peñarol - Hebraica Macabi
Hebraica Macabi - Nacional	Nacional - Goes
Goes - Def. Sporting	Capitol - Def. Sporting
Aguada - Capitol	Olimpia - Aguada
Malvín - Biguá	Trouville - Biguá
FECHA 9	**FECHA 10**
Nacional - Malvín	Capitol - Nacional
Def. Sporting - Trouville	Goes - Urunday U.
Hebraica Macabi - Olimpia	Malvín - Olimpia

Urunday U. - Capitol	Trouville - Hebraica Macabi
Aguada - Goes	Peñarol - Def. Sporting
Biguá - Peñarol	Biguá - Aguada

FECHA 11
Urunday U. - Nacional
Olimpia - Capitol
Goes - Trouville
Def. Sporting - Aguada
Peñarol - Malvín
Hebraica Macabi - Biguá

1.4. Liga Uruguaya de Básquetbol 2021/22

Tabla A1.4: Fixture de la Liga Uruguaya de Básquetbol 2021/22

FECHA 1	**FECHA 2**
Peñarol - Hebraica Macabi	Hebraica Macabi - Malvín
Olimpia - Goes	Aguada - Peñarol
Biguá - Urunday U.	Olivol Mundial - Nacional
Trouville - Nacional	Olimpia - Def. Sporting
Malvín - Aguada	Trouville - Urupán
Def. Sporting - Olivol Mundial	Biguá - Capitol
Urupán - Capitol	Urunday U. - Goes
FECHA 3	**FECHA 4**
Hebraica Macabi - Olimpia	Malvín - Urupán
Goes - Peñarol	Aguada - Def. Sporting
Urunday U. - Trouville	Olivol Mundial - Capitol
Nacional - Biguá	Biguá - Hebraica Macabi
Def. Sporting - Malvín	Peñarol - Urunday U.
Capitol - Aguada	Goes - Trouville
Urupán - Olivol Mundial	Nacional - Olimpia
FECHA 5	**FECHA 6**
Biguá - Malvín	Hebraica Macabi - Def. Sporting
Aguada - Nacional	Urupán - Peñarol
Capitol - Peñarol	Capitol - Goes
Trouville - Hebraica Macabi	Olimpia - Biguá
Def. Sporting - Goes	Urunday U. - Aguada
Olimpia - Olivol Mundial	Malvín - Nacional
Urunday U. - Urupán	Olivol Mundial - Trouville
FECHA 7	**FECHA 8**
Goes - Hebraica Macabi	Hebraica Macabi - Aguada
Peñarol - Olimpia	Peñarol - Olivol Mundial
Nacional - Urunday U.	Nacional - Def. Sporting
Biguá - Trouville	Olimpia - Urupán
Def. Sporting - Urupán	Urunday U. - Malvín
Aguada - Olivol Mundial	Goes - Biguá
Malvín - Capitol	Trouville - Capitol

FECHA 9	FECHA 10
Olivol Mundial - Hebraica Macabi	Olivol Mundial - Malvín
Def. Sporting - Peñarol	Capitol - Def. Sporting
Urupán - Goes	Urupán - Aguada
Urunday U. - Olimpia	Hebraica Macabi - Urunday U.
Aguada - Biguá	Biguá - Peñarol
Capitol - Nacional	Nacional - Goes
Malvín - Trouville	Olimpia - Trouville
FECHA 11	**FECHA 12**
Nacional - Hebraica Macabi	Capitol - Hebraica Macabi
Peñarol - Malvín	Peñarol - Trouville
Goes - Olivol Mundial	Malvín - Goes
Olimpia - Capitol	Aguada - Olimpia
Urunday U. - Def. Sporting	Olivol Mundial - Urunday U.
Biguá - Urupán	Def. Sporting - Biguá
Trouville - Aguada	Urupán - Nacional

FECHA 13
Hebraica Macabi - Urupán
Nacional - Peñarol
Goes - Aguada
Malvín - Olimpia
Capitol - Urunday U.
Olivol Mundial - Biguá
Trouville - Def. Sporting

Apéndice 2

Fixtures obtenidos

2.1. Algoritmo de programación entera

2.1.1. Campeonato Uruguayo de Fútbol

2.1.1.1. Minimización de breaks

Tabla A2.1: Fixture del Campeonato Uruguayo obtenido con el algoritmo de Programación Entera minimizando los breaks

FECHA 1	**FECHA 2**
River Plate - Fénix	Peñarol - Rentistas
Cerrito - Wanderers	Wanderers - Liverpool
Plaza Colonia - Peñarol	Villa Española - Boston River
Rentistas - Villa Española	Mvd. City Torque - Progreso
Progreso - Nacional	Dep. Maldonado - River Plate
Sud América - Dep. Maldonado	Fénix - Cerrito
Liverpool - Cerro Largo	Nacional - Plaza Colonia
Boston River - Mvd. City Torque	Cerro Largo - Sud América
FECHA 3	**FECHA 4**
River Plate - Peñarol	Nacional - Rentistas
Sud América - Fénix	Mvd. City Torque - Liverpool
Liverpool - Villa Española	Dep. Maldonado - Cerrito
Cerrito - Nacional	Cerro Largo - Wanderers
Boston River - Dep. Maldonado	Villa Española - Sud América

Wanderers - Mvd. City Torque	Fénix - Boston River
Progreso - Cerro Largo	Peñarol - Progreso
Rentistas - Plaza Colonia	Plaza Colonia - River Plate
FECHA 5	**FECHA 6**
River Plate - Rentistas	River Plate - Nacional
Progreso - Plaza Colonia	Sud América - Liverpool
Nacional - Dep. Maldonado	Peñarol - Boston River
Wanderers - Villa Española	Dep. Maldonado - Wanderers
Liverpool - Fénix	Plaza Colonia - Cerrito
Mvd. City Torque - Cerro Largo	Fénix - Mvd. City Torque
Cerrito - Peñarol	Villa Española - Cerro Largo
Boston River - Sud América	Rentistas - Progreso
FECHA 7	**FECHA 8**
Wanderers - Sud América	River Plate - Cerro Largo
Liverpool - Plaza Colonia	Sud América - Mvd. City Torque
Mvd. City Torque - Villa Española	Fénix - Villa Española
Cerro Largo - Dep. Maldonado	Peñarol - Liverpool
Boston River - Rentistas	Dep. Maldonado - Progreso
Nacional - Peñarol	Nacional - Boston River
Cerrito - River Plate	Plaza Colonia - Wanderers
Progreso - Fénix	Rentistas - Cerrito
FECHA 9	**FECHA 10**
Wanderers - Rentistas	Dep. Maldonado - Liverpool
Liverpool - Cerrito	Cerrito - Progreso
Villa Española - Dep. Maldonado	Fénix - Wanderers
Progreso - River Plate	Peñarol - Cerro Largo
Sud América - Peñarol	Rentistas - Mvd. City Torque
Boston River - Plaza Colonia	Nacional - Sud América
Mvd. City Torque - Nacional	River Plate - Boston River
Cerro Largo - Fénix	Plaza Colonia - Villa Española
FECHA 11	**FECHA 12**
Mvd. City Torque - Plaza Colonia	Dep. Maldonado - Mvd. City Torque
Sud América - Rentistas	Cerrito - Villa Española
Fénix - Dep. Maldonado	Nacional - Wanderers
Wanderers - Peñarol	River Plate - Sud América
Boston River - Cerrito	Plaza Colonia - Cerro Largo

Liverpool - Progreso	Peñarol - Fénix
Cerro Largo - Nacional	Progreso - Boston River
Villa Española - River Plate	Rentistas - Liverpool
FECHA 13	**FECHA 14**
Liverpool - River Plate	Nacional - Villa Española
Mvd. City Torque - Peñarol	Rentistas - Fénix
Villa Española - Progreso	Boston River - Liverpool
Dep. Maldonado - Plaza Colonia	Peñarol - Dep. Maldonado
Cerro Largo - Rentistas	River Plate - Mvd. City Torque
Wanderers - Boston River	Plaza Colonia - Sud América
Sud América - Cerrito	Cerrito - Cerro Largo
Fénix - Nacional	Progreso - Wanderers

FECHA 15
Liverpool - Nacional
Dep. Maldonado - Rentistas
Fénix - Plaza Colonia
Wanderers - River Plate
Mvd. City Torque - Cerrito
Cerro Largo - Boston River
Villa Española - Peñarol
Sud América - Progreso

2.1.1.2. Minimización de efecto arrastre

Tabla A2.2: Fixture del Campeonato Uruguayo obtenido con el algoritmo de Programación Entera minimizando el efecto arrastre

FECHA 1	**FECHA 2**
Nacional - Mvd. City Torque	Liverpool - Villa Española
Cerrito - Liverpool	Mvd. City Torque - Peñarol
River Plate - Villa Española	River Plate - Plaza Colonia
Dep. Maldonado - Boston River	Dep. Maldonado - Nacional
Cerro Largo - Peñarol	Sud América - Fénix
Plaza Colonia - Fénix	Boston River - Progreso
Wanderers - Progreso	Cerrito - Wanderers
Sud América - Rentistas	Rentistas - Cerro Largo
FECHA 3	**FECHA 4**
River Plate - Cerrito	Wanderers - River Plate
Villa Española - Progreso	Boston River - Peñarol
Fénix - Wanderers	Nacional - Villa Española
Nacional - Cerro Largo	Mvd. City Torque - Rentistas
Liverpool - Sud América	Cerrito - Fénix
Peñarol - Plaza Colonia	Cerro Largo - Liverpool
Mvd. City Torque - Dep. Maldonado	Progreso - Dep. Maldonado
Rentistas - Boston River	Plaza Colonia - Sud América
FECHA 5	**FECHA 6**
Nacional - Fénix	Mvd. City Torque - Cerro Largo
Wanderers - Liverpool	River Plate - Dep. Maldonado
Progreso - Mvd. City Torque	Cerrito - Peñarol
Villa Española - Boston River	Sud América - Progreso
Rentistas - Plaza Colonia	Liverpool - Fénix
Peñarol - River Plate	Boston River - Nacional
Dep. Maldonado - Cerrito	Wanderers - Rentistas
Cerro Largo - Sud América	Villa Española - Plaza Colonia
FECHA 7	**FECHA 8**
Nacional - Peñarol	Plaza Colonia - Dep. Maldonado
Plaza Colonia - Progreso	Fénix - Peñarol
Fénix - Cerro Largo	Cerro Largo - Wanderers
Liverpool - River Plate	Progreso - Rentistas

Villa Española - Rentistas	Nacional - River Plate
Wanderers - Dep. Maldonado	Mvd. City Torque - Sud América
Mvd. City Torque - Boston River	Villa Española - Cerrito
Cerrito - Sud América	Boston River - Liverpool
FECHA 9	**FECHA 10**
Nacional - Liverpool	Cerro Largo - Cerrito
Wanderers - Boston River	Boston River - River Plate
Cerrito - Plaza Colonia	Mvd. City Torque - Villa Española
Cerro Largo - Progreso	Dep. Maldonado - Fénix
Sud América - Villa Española	Progreso - Liverpool
Fénix - Mvd. City Torque	Plaza Colonia - Nacional
River Plate - Rentistas	Sud América - Wanderers
Dep. Maldonado - Peñarol	Rentistas - Peñarol
FECHA 11	**FECHA 12**
Wanderers - Mvd. City Torque	Peñarol - Liverpool
Plaza Colonia - Liverpool	Wanderers - Villa Española
Villa Española - Peñarol	Nacional - Sud América
Fénix - Boston River	Cerrito - Progreso
River Plate - Cerro Largo	Rentistas - Dep. Maldonado
Dep. Maldonado - Sud América	Plaza Colonia - Mvd. City Torque
Progreso - Nacional	Fénix - River Plate
Rentistas - Cerrito	Boston River - Cerro Largo
FECHA 13	**FECHA 14**
Wanderers - Plaza Colonia	Peñarol - Wanderers
Villa Española - Cerro Largo	Villa Española - Dep. Maldonado
Sud América - Boston River	Progreso - Fénix
Fénix - Rentistas	Plaza Colonia - Cerro Largo
Peñarol - Progreso	Liverpool - Mvd. City Torque
Dep. Maldonado - Liverpool	Rentistas - Nacional
Nacional - Cerrito	River Plate - Sud América
River Plate - Mvd. City Torque	Cerrito - Boston River

FECHA 15
Peñarol - Sud América
Dep. Maldonado - Cerro Largo
Nacional - Wanderers

Cerrito - Mvd. City Torque
Fénix - Villa Española
Boston River - Plaza Colonia
Progreso - River Plate
Rentistas - Liverpool

2.1.2. Liga Uruguaya de Básquetbol 2020/21

2.1.2.1. Minimización de breaks

Tabla A2.3: Fixture de la Liga Uruguaya de Básquetbol 2020/21 obtenido con el algoritmo de Programación Entera minimizando los breaks

FECHA 1	**FECHA 2**
Olimpia - Trouville	Aguada - Goes
Nacional - Goes	Def. Sporting - Capitol
Urunday U. - Hebraica Macabi	Olimpia - Urunday U.
Capitol - Malvín	Trouville - Biguá
Peñarol - Aguada	Malvín - Peñarol
Biguá - Def. Sporting	Hebraica Macabi - Nacional
FECHA 3	**FECHA 4**
Peñarol - Trouville	Trouville - Urunday U.
Capitol - Hebraica Macabi	Olimpia - Peñarol
Nacional - Aguada	Def. Sporting - Nacional
Urunday U. - Def. Sporting	Aguada - Capitol
Goes - Malvín	Malvín - Biguá
Biguá - Olimpia	Hebraica Macabi - Goes
FECHA 5	**FECHA 6**
Nacional - Malvín	Aguada - Urunday U.
Goes - Trouville	Def. Sporting - Goes
Urunday U. - Biguá	Trouville - Capitol
Peñarol - Def. Sporting	Malvín - Hebraica Macabi
Hebraica Macabi - Aguada	Biguá - Peñarol
Capitol - Olimpia	Olimpia - Nacional
FECHA 7	**FECHA 8**
Hebraica Macabi - Def. Sporting	Biguá - Nacional
Goes - Olimpia	Urunday U. - Goes
Nacional - Trouville	Def. Sporting - Malvín
Urunday U. - Peñarol	Peñarol - Capitol
Malvín - Aguada	Trouville - Aguada
Capitol - Biguá	Olimpia - Hebraica Macabi
FECHA 9	**FECHA 10**
Aguada - Def. Sporting	Biguá - Aguada

Goes - Biguá	Trouville - Malvín
Malvín - Olimpia	Urunday U. - Nacional
Hebraica Macabi - Trouville	Capitol - Goes
Nacional - Peñarol	Olimpia - Def. Sporting
Capitol - Urunday U.	Peñarol - Hebraica Macabi

FECHA 11
Aguada - Olimpia
Malvín - Urunday U.
Hebraica Macabi - Biguá
Nacional - Capitol
Def. Sporting - Trouville
Goes - Peñarol

2.1.2.2. Minimización de efecto arrastre

Tabla A2.4: Fixture de la Liga Uruguaya de Básquetbol 2020/21 obtenido con el algoritmo de Programación Entera minimizando el efecto arrastre

FECHA 1	**FECHA 2**
Goes - Malvín	Nacional - Aguada
Biguá - Urunday U.	Trouville - Hebraica Macabi
Def. Sporting - Capitol	Olimpia - Def. Sporting
Peñarol - Hebraica Macabi	Urunday U. - Malvín
Olimpia - Aguada	Goes - Peñarol
Trouville - Nacional	Biguá - Capitol
FECHA 3	**FECHA 4**
Hebraica Macabi - Def. Sporting	Urunday U. - Peñarol
Biguá - Nacional	Capitol - Trouville
Peñarol - Olimpia	Malvín - Hebraica Macabi
Trouville - Malvín	Goes - Olimpia
Aguada - Goes	Biguá - Aguada
Urunday U. - Capitol	Def. Sporting - Nacional
FECHA 5	**FECHA 6**
Def. Sporting - Biguá	Goes - Trouville
Trouville - Urunday U.	Nacional - Malvín
Hebraica Macabi - Olimpia	Capitol - Aguada
Aguada - Malvín	Hebraica Macabi - Urunday U.
Goes - Nacional	Biguá - Olimpia
Peñarol - Capitol	Def. Sporting - Peñarol
FECHA 7	**FECHA 8**
Aguada - Def. Sporting	Trouville - Biguá
Goes - Urunday U.	Nacional - Olimpia
Capitol - Olimpia	Urunday U. - Def. Sporting
Malvín - Biguá	Hebraica Macabi - Aguada
Nacional - Hebraica Macabi	Goes - Capitol
Peñarol - Trouville	Peñarol - Malvín
FECHA 9	**FECHA 10**
Hebraica Macabi - Goes	Nacional - Urunday U.
Capitol - Nacional	Goes - Def. Sporting
Aguada - Trouville	Malvín - Capitol

Malvín - Def. Sporting	Biguá - Hebraica Macabi
Urunday U. - Olimpia	Olimpia - Trouville
Peñarol - Biguá	Aguada - Peñarol

FECHA 11
Aguada - Urunday U.
Trouville - Def. Sporting
Malvín - Olimpia
Capitol - Hebraica Macabi
Biguá - Goes
Peñarol - Nacional

2.1.3. Liga Uruguaya de Básquetbol 2021/22

2.1.3.1. Minimización de breaks

Tabla A2.5: Fixture de la Liga Uruguaya de Básquetbol 2021/22 obtenido con el algoritmo de Programación Entera minimizando los breaks

FECHA 1	**FECHA 2**
Olimpia - Peñarol	Biguá - Aguada
Urunday U. - Hebraica Macabi	Hebraica Macabi - Goes
Goes - Trouville	Olivol Mundial - Malvín
Def. Sporting - Biguá	Urupán - Olimpia
Nacional - Olivol Mundial	Trouville - Def. Sporting
Malvín - Urupán	Peñarol - Nacional
Aguada - Capitol	Capitol - Urunday U.
FECHA 3	**FECHA 4**
Hebraica Macabi - Capitol	Biguá - Urunday U.
Goes - Biguá	Olimpia - Malvín
Urunday U. - Trouville	Olivol Mundial - Capitol
Malvín - Peñarol	Urupán - Goes
Aguada - Olivol Mundial	Trouville - Hebraica Macabi
Olimpia - Nacional	Nacional - Def. Sporting
Def. Sporting - Urupán	Peñarol - Aguada
FECHA 5	**FECHA 6**
Hebraica Macabi - Biguá	Biguá - Trouville
Capitol - Trouville	Malvín - Def. Sporting
Malvín - Nacional	Urupán - Aguada
Urunday U. - Urupán	Nacional - Goes
Goes - Olivol Mundial	Olivol Mundial - Hebraica Macabi
Def. Sporting - Peñarol	Olimpia - Urunday U.
Aguada - Olimpia	Peñarol - Capitol
FECHA 7	**FECHA 8**
Capitol - Biguá	Olimpia - Goes
Urunday U. - Peñarol	Malvín - Aguada
Hebraica Macabi - Urupán	Urupán - Capitol
Goes - Malvín	Nacional - Trouville
Aguada - Nacional	Peñarol - Hebraica Macabi

Trouville - Olivol Mundial	Def. Sporting - Urunday U.
Def. Sporting - Olimpia	Olivol Mundial - Biguá
FECHA 9	**FECHA 10**
Urunday U. - Olivol Mundial	Nacional - Urunday U.
Trouville - Malvín	Peñarol - Biguá
Hebraica Macabi - Olimpia	Urupán - Olivol Mundial
Capitol - Nacional	Olimpia - Trouville
Biguá - Urupán	Malvín - Hebraica Macabi
Goes - Peñarol	Def. Sporting - Capitol
Aguada - Def. Sporting	Aguada - Goes
FECHA 11	**FECHA 12**
Biguá - Malvín	Olimpia - Olivol Mundial
Urunday U. - Aguada	Malvín - Urunday U.
Hebraica Macabi - Nacional	Nacional - Biguá
Capitol - Olimpia	Def. Sporting - Hebraica Macabi
Trouville - Urupán	Peñarol - Urupán
Goes - Def. Sporting	Aguada - Trouville
Olivol Mundial - Peñarol	Goes - Capitol

FECHA 13
Olivol Mundial - Def. Sporting
Biguá - Olimpia
Urunday U. - Goes
Trouville - Peñarol
Hebraica Macabi - Aguada
Capitol - Malvín
Urupán - Nacional

2.1.3.2. Minimización de efecto arrastre

Tabla A2.6: Fixture de la Liga Uruguaya de Básquetbol 2021/22 obtenido con el algoritmo de Programación Entera minimizando el efecto arrastre

FECHA 1	**FECHA 2**
Olimpia - Olivol Mundial	Urunday U. - Peñarol
Hebraica Macabi - Nacional	Goes - Def. Sporting
Biguá - Peñarol	Malvín - Nacional
Malvín - Urunday U.	Olivol Mundial - Capitol
Trouville - Aguada	Olimpia - Biguá
Capitol - Def. Sporting	Aguada - Hebraica Macabi
Urupán - Goes	Urupán - Trouville
FECHA 3	**FECHA 4**
Malvín - Olimpia	Malvín - Aguada
Trouville - Nacional	Def. Sporting - Hebraica Macabi
Def. Sporting - Urunday U.	Peñarol - Capitol
Biguá - Urupán	Urupán - Olimpia
Hebraica Macabi - Capitol	Urunday U. - Biguá
Peñarol - Goes	Trouville - Goes
Aguada - Olivol Mundial	Olivol Mundial - Nacional
FECHA 5	**FECHA 6**
Biguá - Def. Sporting	Biguá - Aguada
Olimpia - Goes	Urunday U. - Olivol Mundial
Nacional - Urunday U.	Urupán - Malvín
Aguada - Urupán	Nacional - Peñarol
Trouville - Olivol Mundial	Trouville - Def. Sporting
Hebraica Macabi - Peñarol	Goes - Hebraica Macabi
Capitol - Malvín	Olimpia - Capitol
FECHA 7	**FECHA 8**
Def. Sporting - Peñarol	Trouville - Biguá
Biguá - Olivol Mundial	Capitol - Urunday U.
Trouville - Hebraica Macabi	Olivol Mundial - Urupán
Aguada - Olimpia	Olimpia - Hebraica Macabi
Malvín - Goes	Goes - Aguada
Capitol - Nacional	Def. Sporting - Nacional
Urupán - Urunday U.	Peñarol - Malvín

FECHA 9	FECHA 10
Trouville - Olimpia	Olivol Mundial - Malvín
Aguada - Urunday U.	Nacional - Urupán
Capitol - Biguá	Capitol - Goes
Nacional - Goes	Aguada - Def. Sporting
Malvín - Def. Sporting	Biguá - Hebraica Macabi
Hebraica Macabi - Olivol Mundial	Urunday U. - Trouville
Peñarol - Urupán	Peñarol - Olimpia
FECHA 11	**FECHA 12**
Capitol - Urupán	Olimpia - Nacional
Aguada - Nacional	Urunday U. - Hebraica Macabi
Urunday U. - Olimpia	Trouville - Capitol
Malvín - Hebraica Macabi	Def. Sporting - Urupán
Goes - Biguá	Peñarol - Aguada
Trouville - Peñarol	Olivol Mundial - Goes
Olivol Mundial - Def. Sporting	Malvín - Biguá

FECHA 13
Biguá - Nacional
Urupán - Hebraica Macabi
Trouville - Malvín
Goes - Urunday U.
Aguada - Capitol
Olivol Mundial - Peñarol
Def. Sporting - Olimpia

2.2. Algoritmo evolutivo multi-objetivo

2.2.1. Campeonato Uruguayo de Fútbol

2.2.1.1. Mejor en breaks

Tabla A2.7: Fixture que mejor desempeño obtuvo en breaks de los obtenidos usando el algoritmo evolutivo multi-objetivo para el Campeonato Uruguayo de Fútbol

FECHA 1	**FECHA 2**
Mvd. City Torque - Peñarol	Peñarol - River Plate
Dep. Maldonado - Progreso	Cerro Largo - Dep. Maldonado
Cerrito - Boston River	Progreso - Cerrito
Sud América - Cerro Largo	Boston River - Mvd. City Torque
Plaza Colonia - Villa Española	Villa Española - Sud América
Fénix - Wanderers	Wanderers - Plaza Colonia
Rentistas - Nacional	Nacional - Fénix
River Plate - Liverpool	Liverpool - Rentistas
FECHA 3	**FECHA 4**
Boston River - Peñarol	Peñarol - Rentistas
Dep. Maldonado - Villa Española	Wanderers - Dep. Maldonado
Cerrito - Cerro Largo	Villa Española - Cerrito
Mvd. City Torque - Progreso	Cerro Largo - Mvd. City Torque
Sud América - Wanderers	Progreso - Boston River
Plaza Colonia - Nacional	Nacional - Sud América
Fénix - Liverpool	Liverpool - Plaza Colonia
Rentistas - River Plate	River Plate - Fénix
FECHA 5	**FECHA 6**
Progreso - Peñarol	Peñarol - Fénix
Dep. Maldonado - Nacional	Liverpool - Dep. Maldonado
Cerrito - Wanderers	Nacional - Cerrito
Mvd. City Torque - Villa Española	Wanderers - Mvd. City Torque
Boston River - Cerro Largo	Villa Española - Boston River
Sud América - Liverpool	Cerro Largo - Progreso
Plaza Colonia - River Plate	River Plate - Sud América
Fénix - Rentistas	Rentistas - Plaza Colonia
FECHA 7	**FECHA 8**

Cerro Largo - Peñarol	Peñarol - Plaza Colonia
Dep. Maldonado - River Plate	Rentistas - Dep. Maldonado
Cerrito - Liverpool	River Plate - Cerrito
Mvd. City Torque - Nacional	Liverpool - Mvd. City Torque
Boston River - Wanderers	Nacional - Boston River
Progreso - Villa Española	Wanderers - Progreso
Sud América - Rentistas	Villa Española - Cerro Largo
Plaza Colonia - Fénix	Fénix - Sud América
FECHA 9	**FECHA 10**
Villa Española - Peñarol	Peñarol - Sud América
Dep. Maldonado - Fénix	Plaza Colonia - Dep. Maldonado
Cerrito - Rentistas	Fénix - Cerrito
Mvd. City Torque - River Plate	Rentistas - Mvd. City Torque
Boston River - Liverpool	River Plate - Boston River
Progreso - Nacional	Liverpool - Progreso
Cerro Largo - Wanderers	Nacional - Cerro Largo
Sud América - Plaza Colonia	Wanderers - Villa Española
FECHA 11	**FECHA 12**
Wanderers - Peñarol	Peñarol - Dep. Maldonado
Dep. Maldonado - Sud América	Sud América - Cerrito
Cerrito - Plaza Colonia	Plaza Colonia - Mvd. City Torque
Mvd. City Torque - Fénix	Fénix - Boston River
Boston River - Rentistas	Rentistas - Progreso
Progreso - River Plate	River Plate - Cerro Largo
Cerro Largo - Liverpool	Liverpool - Villa Española
Villa Española - Nacional	Nacional - Wanderers
FECHA 13	**FECHA 14**
Peñarol - Nacional	Cerrito - Peñarol
Cerrito - Dep. Maldonado	Dep. Maldonado - Mvd. City Torque
Mvd. City Torque - Sud América	Sud América - Boston River
Boston River - Plaza Colonia	Plaza Colonia - Progreso
Progreso - Fénix	Fénix - Cerro Largo
Cerro Largo - Rentistas	Rentistas - Villa Española
Villa Española - River Plate	River Plate - Wanderers
Wanderers - Liverpool	Liverpool - Nacional

FECHA 15
Peñarol - Liverpool
Boston River - Dep. Maldonado
Mvd. City Torque - Cerrito
Progreso - Sud América
Cerro Largo - Plaza Colonia
Villa Española - Fénix
Wanderers - Rentistas
Nacional - River Plate

2.2.1.2. Mejor en efecto arrastre

Tabla A2.8: Fixture que mejor desempeño obtuvo en efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para el Campeonato Uruguayo de Fútbol

FECHA 1	**FECHA 2**
River Plate - Villa Española	Villa Española - Cerro Largo
Cerro Largo - Wanderers	Wanderers - Rentistas
Progreso - Nacional	Sud América - Progreso
Rentistas - Liverpool	Liverpool - Plaza Colonia
Boston River - Sud América	Mvd. City Torque - Boston River
Plaza Colonia - Dep. Maldonado	Dep. Maldonado - Peñarol
Cerrito - Mvd. City Torque	Fénix - Cerrito
Peñarol - Fénix	River Plate - Nacional
FECHA 3	**FECHA 4**
Progreso - Villa Española	Villa Española - Mvd. City Torque
Liverpool - Wanderers	Wanderers - Progreso
Boston River - River Plate	River Plate - Liverpool
Cerro Largo - Dep. Maldonado	Cerro Largo - Boston River
Nacional - Cerrito	Nacional - Dep. Maldonado
Rentistas - Fénix	Cerrito - Rentistas
Peñarol - Sud América	Sud América - Fénix
Plaza Colonia - Mvd. City Torque	Plaza Colonia - Peñarol
FECHA 5	**FECHA 6**
Boston River - Villa Española	Villa Española - Rentistas
Fénix - Wanderers	Wanderers - Peñarol
Progreso - Cerrito	Progreso - Fénix
Dep. Maldonado - Liverpool	Liverpool - Cerrito
Peñarol - River Plate	Boston River - Dep. Maldonado
Mvd. City Torque - Cerro Largo	River Plate - Mvd. City Torque
Plaza Colonia - Nacional	Cerro Largo - Plaza Colonia
Rentistas - Sud América	Nacional - Sud América
FECHA 7	**FECHA 8**
Plaza Colonia - Villa Española	Villa Española - Wanderers
Boston River - Wanderers	River Plate - Progreso
Progreso - Liverpool	Liverpool - Cerro Largo

Dep. Maldonado - River Plate	Nacional - Boston River
Cerrito - Cerro Largo	Rentistas - Dep. Maldonado
Fénix - Nacional	Sud América - Cerrito
Peñarol - Rentistas	Fénix - Plaza Colonia
Sud América - Mvd. City Torque	Mvd. City Torque - Peñarol
FECHA 9	**FECHA 10**
Dep. Maldonado - Villa Española	Cerrito - Villa Española
Wanderers - Mvd. City Torque	Sud América - Wanderers
Peñarol - Progreso	Progreso - Plaza Colonia
Liverpool - Fénix	Mvd. City Torque - Liverpool
Cerrito - Boston River	Boston River - Peñarol
Plaza Colonia - River Plate	Fénix - Dep. Maldonado
Cerro Largo - Sud América	River Plate - Rentistas
Rentistas - Nacional	Nacional - Cerro Largo
FECHA 11	**FECHA 12**
Villa Española - Sud América	Villa Española - Peñarol
Wanderers - Cerrito	River Plate - Wanderers
Dep. Maldonado - Progreso	Progreso - Cerro Largo
Liverpool - Boston River	Nacional - Liverpool
Fénix - River Plate	Boston River - Rentistas
Cerro Largo - Peñarol	Sud América - Dep. Maldonado
Mvd. City Torque - Nacional	Cerrito - Plaza Colonia
Rentistas - Plaza Colonia	Mvd. City Torque - Fénix
FECHA 13	**FECHA 14**
Fénix - Villa Española	Villa Española - Liverpool
Wanderers - Nacional	Wanderers - Dep. Maldonado
Rentistas - Progreso	Boston River - Progreso
Liverpool - Sud América	Cerrito - River Plate
Plaza Colonia - Boston River	Cerro Largo - Fénix
Dep. Maldonado - Mvd. City Torque	Peñarol - Nacional
Peñarol - Cerrito	Mvd. City Torque - Rentistas
Cerro Largo - River Plate	Sud América - Plaza Colonia

FECHA 15
Nacional - Villa Española
Plaza Colonia - Wanderers

Progreso - Mvd. City Torque
Liverpool - Peñarol
Fénix - Boston River
Dep. Maldonado - Cerrito
River Plate - Sud América
Rentistas - Cerro Largo

2.2.1.3. Balanceado en breaks y efecto arrastre

Tabla A2.9: Fixture que obtuvo un desempeño balanceado en breaks y efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para el Campeonato Uruguayo de Fútbol

FECHA 1	**FECHA 2**
Progreso - Cerrito	Cerrito - Rentistas
Sud América - Dep. Maldonado	Nacional - Sud América
Plaza Colonia - Nacional	Liverpool - Plaza Colonia
Fénix - Liverpool	Progreso - Fénix
Boston River - Cerro Largo	Dep. Maldonado - Boston River
Peñarol - Villa Española	Cerro Largo - Peñarol
Wanderers - Mvd. City Torque	Villa Española - Wanderers
River Plate - Rentistas	Mvd. City Torque - River Plate
FECHA 3	**FECHA 4**
Fénix - Cerrito	Cerrito - River Plate
Sud América - Liverpool	Cerro Largo - Sud América
Plaza Colonia - Progreso	Dep. Maldonado - Plaza Colonia
Boston River - Nacional	Nacional - Fénix
Peñarol - Dep. Maldonado	Liverpool - Progreso
Wanderers - Cerro Largo	Villa Española - Boston River
River Plate - Villa Española	Mvd. City Torque - Peñarol
Rentistas - Mvd. City Torque	Rentistas - Wanderers
FECHA 5	**FECHA 6**
Liverpool - Cerrito	Cerrito - Wanderers
Sud América - Villa Española	Mvd. City Torque - Sud América
Plaza Colonia - Cerro Largo	Villa Española - Plaza Colonia
Fénix - Dep. Maldonado	Cerro Largo - Fénix
Progreso - Nacional	Dep. Maldonado - Progreso
Boston River - Mvd. City Torque	Nacional - Liverpool
Peñarol - Rentistas	Rentistas - Boston River
Wanderers - River Plate	River Plate - Peñarol
FECHA 7	**FECHA 8**
Cerrito - Nacional	Peñarol - Cerrito
Sud América - Rentistas	River Plate - Sud América
Plaza Colonia - Mvd. City Torque	Rentistas - Plaza Colonia

Fénix - Villa Española	Mvd. City Torque - Fénix
Progreso - Cerro Largo	Villa Española - Progreso
Liverpool - Dep. Maldonado	Cerro Largo - Liverpool
Boston River - River Plate	Dep. Maldonado - Nacional
Wanderers - Peñarol	Wanderers - Boston River
FECHA 9	**FECHA 10**
Dep. Maldonado - Cerrito	Cerrito - Plaza Colonia
Sud América - Wanderers	Sud América - Fénix
Plaza Colonia - River Plate	Boston River - Progreso
Fénix - Rentistas	Peñarol - Liverpool
Progreso - Mvd. City Torque	Wanderers - Nacional
Liverpool - Villa Española	River Plate - Dep. Maldonado
Nacional - Cerro Largo	Rentistas - Cerro Largo
Boston River - Peñarol	Mvd. City Torque - Villa Española
FECHA 11	**FECHA 12**
Cerrito - Cerro Largo	Boston River - Cerrito
Sud América - Boston River	Peñarol - Sud América
Plaza Colonia - Peñarol	Wanderers - Plaza Colonia
Fénix - Wanderers	River Plate - Fénix
Progreso - River Plate	Rentistas - Progreso
Liverpool - Rentistas	Mvd. City Torque - Liverpool
Nacional - Mvd. City Torque	Villa Española - Nacional
Dep. Maldonado - Villa Española	Cerro Largo - Dep. Maldonado
FECHA 13	**FECHA 14**
Cerrito - Villa Española	Sud América - Cerrito
Plaza Colonia - Sud América	Boston River - Plaza Colonia
Fénix - Boston River	Peñarol - Fénix
Progreso - Peñarol	Wanderers - Progreso
Liverpool - Wanderers	River Plate - Liverpool
Nacional - River Plate	Rentistas - Nacional
Dep. Maldonado - Rentistas	Mvd. City Torque - Dep. Maldonado
Cerro Largo - Mvd. City Torque	Villa Española - Cerro Largo

FECHA 15
Cerrito - Mvd. City Torque
Progreso - Sud América

Fénix - Plaza Colonia
Liverpool - Boston River
Nacional - Peñarol
Dep. Maldonado - Wanderers
Cerro Largo - River Plate
Villa Española - Rentistas

2.2.2. Liga Uruguaya de Básquetbol 2020/21

2.2.2.1. Mejor en breaks

Tabla A2.10: Fixture que mejor desempeño obtuvo en breaks de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2020/21

FECHA 1	**FECHA 2**
Malvín - Def. Sporting	Olimpia - Malvín
Peñarol - Trouville	Trouville - Urunday U.
Urunday U. - Olimpia	Biguá - Hebraica Macabi
Hebraica Macabi - Capitol	Def. Sporting - Goes
Goes - Aguada	Aguada - Nacional
Nacional - Biguá	Capitol - Peñarol
FECHA 3	**FECHA 4**
Malvín - Aguada	Urunday U. - Malvín
Nacional - Trouville	Trouville - Hebraica Macabi
Peñarol - Urunday U.	Biguá - Goes
Hebraica Macabi - Olimpia	Aguada - Def. Sporting
Goes - Capitol	Capitol - Nacional
Def. Sporting - Biguá	Olimpia - Peñarol
FECHA 5	**FECHA 6**
Malvín - Biguá	Peñarol - Malvín
Def. Sporting - Trouville	Trouville - Goes
Nacional - Urunday U.	Urunday U. - Hebraica Macabi
Hebraica Macabi - Peñarol	Biguá - Aguada
Goes - Olimpia	Capitol - Def. Sporting
Aguada - Capitol	Olimpia - Nacional
FECHA 7	**FECHA 8**
Malvín - Capitol	Hebraica Macabi - Malvín
Biguá - Trouville	Trouville - Aguada
Def. Sporting - Urunday U.	Urunday U. - Goes
Nacional - Hebraica Macabi	Capitol - Biguá
Goes - Peñarol	Olimpia - Def. Sporting
Aguada - Olimpia	Peñarol - Nacional
FECHA 9	**FECHA 10**

Malvín - Trouville	Nacional - Malvín
Biguá - Urunday U.	Trouville - Capitol
Def. Sporting - Hebraica Macabi	Urunday U. - Aguada
Goes - Nacional	Hebraica Macabi - Goes
Aguada - Peñarol	Olimpia - Biguá
Capitol - Olimpia	Peñarol - Def. Sporting

FECHA 11
Goes - Malvín
Olimpia - Trouville
Capitol - Urunday U.
Aguada - Hebraica Macabi
Biguá - Peñarol
Def. Sporting - Nacional

2.2.2.2. Mejor en efecto arrastre

Tabla A2.11: Fixture que mejor desempeño obtuvo en efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2020/21

FECHA 1	**FECHA 2**
Trouville - Urunday U.	Hebraica Macabi - Trouville
Aguada - Goes	Capitol - Aguada
Nacional - Def. Sporting	Def. Sporting - Biguá
Peñarol - Olimpia	Urunday U. - Peñarol
Biguá - Hebraica Macabi	Goes - Nacional
Malvín - Capitol	Olimpia - Malvín
FECHA 3	**FECHA 4**
Peñarol - Trouville	Trouville - Biguá
Olimpia - Aguada	Aguada - Def. Sporting
Def. Sporting - Hebraica Macabi	Urunday U. - Olimpia
Malvín - Urunday U.	Hebraica Macabi - Nacional
Nacional - Capitol	Goes - Malvín
Biguá - Goes	Capitol - Peñarol
FECHA 5	**FECHA 6**
Malvín - Trouville	Trouville - Goes
Peñarol - Aguada	Aguada - Malvín
Def. Sporting - Goes	Capitol - Def. Sporting
Nacional - Urunday U.	Urunday U. - Biguá
Biguá - Olimpia	Peñarol - Nacional
Hebraica Macabi - Capitol	Olimpia - Hebraica Macabi
FECHA 7	**FECHA 8**
Nacional - Trouville	Trouville - Capitol
Biguá - Aguada	Aguada - Nacional
Def. Sporting - Peñarol	Urunday U. - Def. Sporting
Goes - Urunday U.	Olimpia - Goes
Capitol - Olimpia	Peñarol - Hebraica Macabi
Hebraica Macabi - Malvín	Malvín - Biguá
FECHA 9	**FECHA 10**
Aguada - Trouville	Trouville - Def. Sporting
Def. Sporting - Olimpia	Urunday U. - Aguada

Hebraica Macabi - Urunday U.	Olimpia - Nacional
Nacional - Malvín	Goes - Hebraica Macabi
Goes - Capitol	Malvín - Peñarol
Biguá - Peñarol	Capitol - Biguá

FECHA 11
Trouville - Olimpia
Aguada - Hebraica Macabi
Def. Sporting - Malvín
Urunday U. - Capitol
Biguá - Nacional
Peñarol - Goes

2.2.2.3. Balanceado en breaks y efecto arrastre

Tabla A2.12: Fixture que obtuvo un desempeño balanceado en breaks y efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2020/21

FECHA 1	**FECHA 2**
Hebraica Macabi - Urunday U.	Def. Sporting - Hebraica Macabi
Capitol - Goes	Goes - Peñarol
Trouville - Peñarol	Nacional - Olimpia
Olimpia - Def. Sporting	Malvín - Aguada
Aguada - Biguá	Urunday U. - Capitol
Malvín - Nacional	Biguá - Trouville
FECHA 3	**FECHA 4**
Hebraica Macabi - Malvín	Peñarol - Hebraica Macabi
Trouville - Goes	Goes - Olimpia
Peñarol - Def. Sporting	Nacional - Aguada
Olimpia - Biguá	Urunday U. - Malvín
Aguada - Urunday U.	Biguá - Capitol
Capitol - Nacional	Def. Sporting - Trouville
FECHA 5	**FECHA 6**
Hebraica Macabi - Nacional	Trouville - Hebraica Macabi
Malvín - Goes	Goes - Aguada
Capitol - Peñarol	Peñarol - Olimpia
Olimpia - Trouville	Nacional - Urunday U.
Aguada - Def. Sporting	Biguá - Malvín
Urunday U. - Biguá	Def. Sporting - Capitol
FECHA 7	**FECHA 8**
Hebraica Macabi - Biguá	Olimpia - Hebraica Macabi
Nacional - Goes	Goes - Urunday U.
Malvín - Peñarol	Peñarol - Aguada
Capitol - Olimpia	Biguá - Nacional
Aguada - Trouville	Def. Sporting - Malvín
Urunday U. - Def. Sporting	Trouville - Capitol
FECHA 9	**FECHA 10**
Hebraica Macabi - Goes	Capitol - Hebraica Macabi
Nacional - Peñarol	Goes - Biguá

Malvín - Olimpia	Peñarol - Urunday U.
Aguada - Capitol	Olimpia - Aguada
Urunday U. - Trouville	Def. Sporting - Nacional
Biguá - Def. Sporting	Trouville - Malvín

FECHA 11
Hebraica Macabi - Aguada
Def. Sporting - Goes
Biguá - Peñarol
Urunday U. - Olimpia
Nacional - Trouville
Malvín - Capitol

2.2.3. Liga Uruguaya de Básquetbol 2021/22

2.2.3.1. Mejor en breaks

Tabla A2.13: Fixture que mejor desempeño obtuvo en breaks de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2021/22

FECHA 1	**FECHA 2**
Olimpia - Urunday U.	Urunday U. - Biguá
Def. Sporting - Hebraica Macabi	Nacional - Def. Sporting
Biguá - Trouville	Trouville - Olivol Mundial
Olivol Mundial - Urupán	Hebraica Macabi - Olimpia
Peñarol - Nacional	Urupán - Capitol
Capitol - Aguada	Goes - Peñarol
Malvín - Goes	Aguada - Malvín
FECHA 3	**FECHA 4**
Urunday U. - Trouville	Urupán - Urunday U.
Def. Sporting - Olimpia	Def. Sporting - Peñarol
Biguá - Urupán	Aguada - Trouville
Peñarol - Hebraica Macabi	Olimpia - Malvín
Olivol Mundial - Aguada	Goes - Biguá
Malvín - Nacional	Hebraica Macabi - Capitol
Capitol - Goes	Nacional - Olivol Mundial
FECHA 5	**FECHA 6**
Urunday U. - Def. Sporting	Peñarol - Urunday U.
Trouville - Urupán	Def. Sporting - Malvín
Peñarol - Olimpia	Goes - Trouville
Biguá - Aguada	Olimpia - Capitol
Malvín - Hebraica Macabi	Nacional - Biguá
Olivol Mundial - Goes	Hebraica Macabi - Olivol Mundial
Capitol - Nacional	Aguada - Urupán
FECHA 7	**FECHA 8**
Urunday U. - Aguada	Goes - Urunday U.
Capitol - Def. Sporting	Def. Sporting - Biguá
Trouville - Nacional	Olimpia - Trouville
Olivol Mundial - Olimpia	Hebraica Macabi - Urupán

Biguá - Hebraica Macabi	Peñarol - Olivol Mundial
Urupán - Goes	Nacional - Aguada
Malvín - Peñarol	Malvín - Capitol
FECHA 9	**FECHA 10**
Urunday U. - Malvín	Nacional - Urunday U.
Olivol Mundial - Def. Sporting	Def. Sporting - Urupán
Trouville - Hebraica Macabi	Peñarol - Trouville
Biguá - Olimpia	Olimpia - Aguada
Urupán - Nacional	Malvín - Biguá
Capitol - Peñarol	Hebraica Macabi - Goes
Aguada - Goes	Capitol - Olivol Mundial
FECHA 11	**FECHA 12**
Urunday U. - Capitol	Olivol Mundial - Urunday U.
Trouville - Def. Sporting	Def. Sporting - Aguada
Urupán - Olimpia	Malvín - Trouville
Biguá - Peñarol	Olimpia - Goes
Aguada - Hebraica Macabi	Capitol - Biguá
Olivol Mundial - Malvín	Hebraica Macabi - Nacional
Goes - Nacional	Peñarol - Urupán

FECHA 13
Urunday U. - Hebraica Macabi
Goes - Def. Sporting
Trouville - Capitol
Nacional - Olimpia
Biguá - Olivol Mundial
Urupán - Malvín
Aguada - Peñarol

2.2.3.2. Mejor en efecto arrastre

Tabla A2.14: Fixture que mejor desempeño obtuvo en efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2021/22

FECHA 1	**FECHA 2**
Aguada - Peñarol	Trouville - Aguada
Capitol - Olimpia	Olimpia - Olivol Mundial
Hebraica Macabi - Def. Sporting	Hebraica Macabi - Goes
Urunday U. - Malvín	Malvín - Nacional
Olivol Mundial - Trouville	Peñarol - Urunday U.
Goes - Biguá	Def. Sporting - Urupán
Nacional - Urupán	Capitol - Biguá
FECHA 3	**FECHA 4**
Aguada - Goes	Biguá - Aguada
Olimpia - Peñarol	Hebraica Macabi - Olimpia
Urupán - Hebraica Macabi	Malvín - Olivol Mundial
Biguá - Malvín	Goes - Urunday U.
Urunday U. - Trouville	Nacional - Def. Sporting
Def. Sporting - Capitol	Capitol - Peñarol
Olivol Mundial - Nacional	Trouville - Urupán
FECHA 5	**FECHA 6**
Aguada - Capitol	Hebraica Macabi - Aguada
Olimpia - Nacional	Malvín - Olimpia
Peñarol - Hebraica Macabi	Urunday U. - Olivol Mundial
Urupán - Malvín	Goes - Def. Sporting
Urunday U. - Biguá	Nacional - Capitol
Def. Sporting - Trouville	Trouville - Peñarol
Olivol Mundial - Goes	Biguá - Urupán
FECHA 7	**FECHA 8**
Olivol Mundial - Aguada	Aguada - Def. Sporting
Olimpia - Goes	Urupán - Olimpia
Nacional - Hebraica Macabi	Biguá - Hebraica Macabi
Malvín - Peñarol	Trouville - Malvín
Urupán - Urunday U.	Urunday U. - Capitol
Def. Sporting - Biguá	Peñarol - Olivol Mundial

Capitol - Trouville	Goes - Nacional
FECHA 9	**FECHA 10**
Aguada - Urupán	Malvín - Aguada
Olimpia - Urunday U.	Def. Sporting - Olimpia
Hebraica Macabi - Malvín	Urunday U. - Hebraica Macabi
Def. Sporting - Olivol Mundial	Olivol Mundial - Capitol
Capitol - Goes	Goes - Trouville
Nacional - Trouville	Biguá - Nacional
Peñarol - Biguá	Urupán - Peñarol
FECHA 11	**FECHA 12**
Aguada - Olimpia	Urunday U. - Aguada
Hebraica Macabi - Olivol Mundial	Olimpia - Trouville
Goes - Malvín	Hebraica Macabi - Capitol
Nacional - Urunday U.	Malvín - Def. Sporting
Peñarol - Def. Sporting	Olivol Mundial - Biguá
Capitol - Urupán	Urupán - Goes
Trouville - Biguá	Peñarol - Nacional

FECHA 13
Nacional - Aguada
Biguá - Olimpia
Trouville - Hebraica Macabi
Capitol - Malvín
Def. Sporting - Urunday U.
Olivol Mundial - Urupán
Goes - Peñarol

2.2.3.3. Balanceado en breaks y efecto arrastre

Tabla A2.15: Fixture que obtuvo un desempeño balanceado en breaks y efecto arrastre de los obtenidos usando el algoritmo evolutivo multi-objetivo para la Liga Uruguaya de Básquetbol 2021/22

FECHA 1	**FECHA 2**
Olimpia - Peñarol	Olimpia - Biguá
Olivol Mundial - Nacional	Urunday U. - Olivol Mundial
Biguá - Urupán	Peñarol - Capitol
Goes - Hebraica Macabi	Hebraica Macabi - Nacional
Capitol - Trouville	Urupán - Aguada
Def. Sporting - Urunday U.	Malvín - Goes
Aguada - Malvín	Trouville - Def. Sporting
FECHA 3	**FECHA 4**
Nacional - Olimpia	Urupán - Olimpia
Olivol Mundial - Hebraica Macabi	Olivol Mundial - Goes
Biguá - Peñarol	Trouville - Peñarol
Capitol - Urupán	Nacional - Def. Sporting
Goes - Urunday U.	Malvín - Biguá
Aguada - Trouville	Hebraica Macabi - Aguada
Def. Sporting - Malvín	Urunday U. - Capitol
FECHA 5	**FECHA 6**
Olimpia - Olivol Mundial	Malvín - Olimpia
Peñarol - Urupán	Olivol Mundial - Biguá
Goes - Nacional	Nacional - Peñarol
Biguá - Trouville	Hebraica Macabi - Urupán
Def. Sporting - Hebraica Macabi	Goes - Capitol
Capitol - Malvín	Urunday U. - Trouville
Aguada - Urunday U.	Def. Sporting - Aguada
FECHA 7	**FECHA 8**
Olimpia - Trouville	Goes - Olimpia
Aguada - Olivol Mundial	Olivol Mundial - Def. Sporting
Peñarol - Urunday U.	Malvín - Peñarol
Capitol - Nacional	Nacional - Aguada
Biguá - Hebraica Macabi	Urunday U. - Biguá
Urupán - Malvín	Hebraica Macabi - Capitol

Def. Sporting - Goes	Trouville - Urupán
FECHA 9	**FECHA 10**
Olimpia - Def. Sporting	Urunday U. - Olimpia
Capitol - Olivol Mundial	Olivol Mundial - Urupán
Peñarol - Hebraica Macabi	Goes - Peñarol
Biguá - Nacional	Nacional - Trouville
Urupán - Urunday U.	Def. Sporting - Biguá
Aguada - Goes	Hebraica Macabi - Malvín
Trouville - Malvín	Aguada - Capitol
FECHA 11	**FECHA 12**
Olimpia - Aguada	Capitol - Olimpia
Peñarol - Olivol Mundial	Olivol Mundial - Trouville
Urupán - Nacional	Def. Sporting - Peñarol
Biguá - Goes	Nacional - Malvín
Trouville - Hebraica Macabi	Aguada - Biguá
Capitol - Def. Sporting	Hebraica Macabi - Urunday U.
Malvín - Urunday U.	Goes - Urupán

FECHA 13
Olimpia - Hebraica Macabi
Malvín - Olivol Mundial
Peñarol - Aguada
Urunday U. - Nacional
Biguá - Capitol
Urupán - Def. Sporting
Trouville - Goes