Universidad de la República

Facultad de Ingeniería

Confección del fixture para competencias artísticas: una aplicación al Carnaval uruguayo

Proyecto de Grado - Ingeniería de Producción

Autores:

Esteban Machado

Inés Baráibar

Tutores:

Ing. Pedro Piñeyro

Ing. Héctor Cancela

Octubre 2021

Montevideo, Uruguay

Resumen

Este informe presenta el trabajo realizado sobre la Planificación de espectáculos artísticos mediante Programación con Restricciones. Nuestro caso de aplicación se basó en diseñar el calendario correspondiente al Carnaval de las Promesas con el objetivo de equilibrar la concurrencia del público en las diferentes etapas del mismo.

Se consideró para el modelado del problema el reglamento del concurso y las restricciones particulares indicadas por la organización, conformada por la Intendencia de Montevideo y la Asociación de Directores del Carnaval de las Promesas (ADICAPRO).

El Carnaval es la mayor fiesta popular uruguaya y es considerado como el más largo del mundo, con una duración de aproximadamente 40 días. En este marco, es que se realiza el Carnaval de las Promesas, participando de este alrededor de 40 agrupaciones de niños y jóvenes de hasta 18 años de edad, en las categorías murgas, revistas, lubolos, parodistas y humoristas.

La Programación con Restricciones es una técnica de optimización matemática dentro de la programación informática. Es utilizada para describir y resolver diversos problemas en diferentes áreas, como la investigación operativa, inteligencia artificial, base de datos, etc. Aplicándose a diversos problemas de asignación de recursos, toma de decisiones, planificación, entre otros. Consiste en modelar el problema mediante la declaración de restricciones y encontrar soluciones que satisfagan todas las restricciones.

Se relevó y analizó la literatura en torno a problemas de planificación de espectáculos artísticos y eventos en general mediante Programación con Restricciones. Se consideró además el análisis realizado en un proyecto final de la carrera Ingeniería de Producción, en el cual se diseñó el calendario del Carnaval de las Promesas del año 2020, aunque en este caso mediante Programación Lineal.

Debido a la necesidad imperiosa por parte de ADICAPRO de obtener el calendario para el Carnaval de las Promesas 2021 se consideró en la primera etapa continuar con el modelo matemático desarrollado en el Proyecto antes mencionado realizando una revisión del mismo. Luego se realiza el modelado mediante Programación con Restricciones, esto permitió realizar una comparativa de las dos técnicas.

Es importante destacar que este proyecto se realizó en el marco de la pandemia por COVID-19. La incertidumbre que se manejaba por desconocimiento de la misma y la situación del país debido a esta, tuvo gran incidencia y fue un factor relevante a la hora de diseñar los modelos matemáticos y llevar a cabo este trabajo.

Si bien finalmente la edición del Carnaval de las promesas 2021 no se llevó a cabo debido a la situación del país frente a la pandemia, se considera que los objetivos del proyecto fueron alcanzados. Se lograron modelar los calendarios mediante las dos técnicas de optimización matemática, Programación Lineal Entera y Programación con Restricciones pudiendo validar con ADICAPRO las distintas alternativas de fixture y realizar un comparativa entre los resultados de la experimentación de ambas técnicas.

Palabras clave: Programación con Restricciones, Fixture, Planificación de Eventos, Optimización, Timetabling.

Agradecimientos

Queremos agradecer a todas aquellas personas que de alguna forma u otra formaron parte de este proyecto, especialmente en el marco en el que se llevó a cabo debido a la pandemia por COVID-19.

A nuestros tutores Pedro Piñeyro y Hector Cancela, por permitirnos participar de este proyecto, brindarnos su experiencia, conocimiento y visión a lo largo del camino recorrido. A nuestros compañeros de carrera Alfonsina Cardozo, Carolina Guido y Juan Carlos Machín quienes nos transmitieron su experiencia en su Proyecto final de carrera y nos motivaron a continuarlo.

A la Gerencia de Eventos de la Intendencia de Montevideo en conjunto con la Asociación de Directores del Carnaval de las Promesas (ADICAPRO) quienes estuvieron a disposición en todo momento. En este sentido, un agradecimiento especial a Ángel Duarte y Betty Cuccaro.

Agradecemos a nuestros familiares y amigos por su apoyo incondicional.

ÍNDICE

Índice de Figuras

Índice de Tablas

1. Introducción

Este documento presenta el trabajo realizado sobre la planificación de espectáculos artísticos mediante Programación con Restricciones. Nuestro caso de aplicación se basó en diseñar el calendario correspondiente al Carnaval de las Promesas.

La motivación por la que se lleva a cabo este proyecto radica en la complejidad de confeccionar la planificación de competencias artísticas debido a la cantidad de restricciones que ésta puede implicar y la subjetividad que acarrea como consecuencia de la confección manual. Poder aplicar el conocimiento adquirido a lo largo de la carrera y colaborar con el Carnaval de las Promesas, el cual es valorado por la sociedad como un evento cultural importante para la cultura local, se considera sumamente enriquecedor.

Resulta interesante abordar la planificación de eventos. En este sentido, un evento se define como “un suceso importante y programado, de índole social, académica, artística o deportiva” [1]. La programación de eventos refiere a la coordinación y planificación de estos eventos asignando el comienzo y finalización de cada uno de ellos considerando las restricciones propias de cada evento y otros factores que pueden tener que ver con la organización, reglamentaciones, términos económicos entre otros.

Dentro de la planificación de eventos deportivos, la planificación de las ligas deportivas es uno de los problemas más desafiantes; como por ejemplo, planificar el fixture de una liga asignando fechas, partidos, localías, giras entre otros factores.

En el ámbito académico la planificación de horarios de clase o exámenes, en el ámbito social y cultural la planificación de competencias, eventos o espectáculos. Cabe destacar que, en este rubro, el fin muchas veces no es solamente económico, siendo que incluso este factor directamente no es considerado cuando el mismo tiene como objetivo el bienestar social.

El Carnaval es la mayor fiesta popular uruguaya y es considerado como el más largo del mundo, con una duración de aproximadamente 40 días. En este marco, es que se realiza el Carnaval de las Promesas, participando de este alrededor de 40 agrupaciones de niños y jóvenes de hasta 18 años de edad, en las categorías murgas, revistas, lubolos, parodistas y humoristas.

La Programación con Restricciones es una técnica de optimización matemática dentro de la programación informática. Es utilizada para describir y resolver diversos problemas en diferentes áreas, como la investigación operativa, inteligencia artificial, base de datos, etc. Aplicándose a diversos problemas de asignación de recursos, toma de decisiones, planificación, entre otros. Consiste en modelar el problema mediante la declaración de restricciones y encontrar soluciones que satisfagan todas las restricciones.

Este caso de aplicación se basó en diseñar el calendario correspondiente al Carnaval de las Promesas con el objetivo de equilibrar la concurrencia del público en las diferentes etapas del mismo. Se consideró para el modelado del problema el reglamento del concurso y las restricciones particulares indicadas por la organización, conformada por la Intendencia de Montevideo y la Asociación de Directores del Carnaval de las Promesas (ADICAPRO).

Dada la complejidad de la problemática, la resolución de forma manual se vuelve compleja, por lo tanto es necesario recurrir a técnicas de optimización matemática y solvers, (solver o solucionador es una pieza de software matemático, posiblemente en forma de un programa de computadora independiente o como una biblioteca de software, que "resuelve" un problema

matemático [2]), que permitan abordar estos problemas y obtener soluciones óptimas o cercanas a las óptimas de forma rápida en términos de tiempos de ejecución. Dentro de las técnicas de optimización matemática se encuentran la Programación Matemática, (Programación Lineal, Programación Entera, Programación Entera-Mixta entre otras), y la Programación con Restricciones.

El objetivo principal del proyecto se divide en dos etapas. Debido a la necesidad imperiosa de ADICAPRO de obtener un calendario para el Carnaval de las Promesas 2021, se consideró en la primera etapa continuar con el modelo matemático desarrollado en el proyecto previamente mencionado, realizando una revisión del modelo. La revisión del mismo implicó también adaptar parte del modelo matemático a las características propias de esta edición. En particular, es importante destacar que este proyecto se realizó en el marco de la pandemia por COVID-19. La incertidumbre que se manejaba por desconocimiento de la misma y la situación del país debido a esta, tuvo gran incidencia y fue un factor relevante a la hora de diseñar los modelos matemáticos.

Luego en una segunda etapa se realizó el modelado mediante Programación con Restricciones. Esto requirió un estudio previo de esta técnica, así como también la selección del solver para implementar los modelos, lo cual permitió realizar una comparativa de las dos técnicas.

Para poder llevar a cabo la segunda etapa, se realizó el relevamiento y análisis de la literatura en torno a problemas de planificación mediante Programación con Restricciones. Se consideró el análisis realizado en un proyecto final de la carrera Ingeniería de Producción, en el cual también se diseñó el calendario del Carnaval de las Promesas del año 2020, aunque en este caso, utilizando Programación Lineal Entera. Este relevamiento incorpora un breve análisis sobre las herramientas utilizadas para resolver problemas de este tipo.

Si bien finalmente la edición del Carnaval de las promesas 2021 no se llevó a cabo debido a la situación del país frente a la pandemia, se considera que los objetivos fueron alcanzados. Se logró un importante relevamiento sobre la temática de este proyecto, así como modelar los calendarios mediante las dos técnicas de optimización matemática, (Programación Lineal Entera y Programación con Restricciones). A su vez, se logró validar con ADICAPRO las distintas alternativas de fixture y realizar una comparativa entre los resultados de la experimentación de ambas técnicas.

El resto del documento está organizado de la siguiente forma: en la Sección 2 se presenta el marco teórico del caso de estudio; en la Sección 3 se desarrolla el caso de estudio; en la Sección 4 la Experimentación numérica y el análisis comparativo; en la Sección 5 se presentan las conclusiones y sugerencias para investigaciones a futuro surgidas en este trabajo.

2. Marco Teórico

En la presente sección se describe una introducción a la Planificación de eventos y Programación con Restricciones. En el Estado del Arte, (Anexo 1), se tratan con mayor profundidad.

A continuación, el documento se desarrolla de la siguiente manera: en la Sección 2.1 Planificación de eventos, en la Sección 2.2 Técnicas de optimización matemática aplicada a Planificación de eventos. Por último, en 2.3 se estudia el objeto de investigación de este proyecto, el Carnaval de las Promesas, y su relación con el Carnaval uruguayo.

2.1. Planificación de eventos

“Se conoce como planificación, planeación o planteamiento, al proceso de toma de decisiones para alcanzar un futuro deseado, teniendo en cuenta la situación actual y los factores internos y externos que pueden influir en el logro de los objetivos” [3].

Planificar implica analizar y estudiar los objetivos propuestos, así como la forma en que se alcanzarán [4]. Es una herramienta que permite decidir qué se va a hacer y por qué. Planificar supone muchos beneficios entre los cuales se cuentan clarificar dudas, ayuda a determinar la necesidad de recursos para lograr los objetivos, ayuda a estructurar las actividades y clarificar las dudas respecto a los objetivos buscados, cuantificar los niveles de desempeño para tener éxito, establecer prioridades, y evidenciar debilidades y fortalezas para conseguir los objetivos propuestos.

Se entiende a su vez como una actividad humana esencial para la supervivencia de una comunidad, la cual tenemos incorporada intrínsecamente en nuestra vida cotidiana para lograr nuestros objetivos.

A efectos de este estudio se profundizó en problemas de planificación en los cuales se aplican técnicas de optimización matemática como ser Programación Matemática y Programación con Restricciones: planificación de eventos sociales y culturales y, por otro lado, deportivos. La mayoría de los trabajos encontrados son relativos al deporte: basquetbol, fútbol y voley. Esto se explica debido a que este tipo de eventos involucran grandes cantidades de dinero, de ahí la importancia de optimizar tiempos y recursos. También se relevaron artículos relativos a eventos académicos, (programación de clases o exámenes), avisos publicitarios entre otros.

2.2. Técnicas de optimización matemática aplicado a Planificación de eventos

Dentro de los eventos deportivos, las técnicas de optimización matemática son ampliamente utilizadas. Éstas permiten confeccionar calendarios para torneos: se han utilizado en básquetbol, volleyball, fútbol, entre otros, [5], [6], [7]. En [5] se aplica Programación Lineal Entera mientras que en [6] y [7] se utiliza Programación con Restricciones. Entre las ventajas que se obtienen al utilizar estas técnicas se cuentan: obtener fixtures de forma objetiva con criterios claros, ahorro de tiempo y dinero y evitar los errores que se podrían cometer si el calendario se confeccionara de forma manual. También, dentro del rubro deportivo, se encuentra en [8] el problema de la asignación de voluntarios a un evento deportivo de gran dimensión, en donde se toma como caso

de estudio la XVII Copa Mundial de Baloncesto FIBA que tuvo lugar en España en el año 2014. En este último caso también se aplica Programación Lineal Entera.

En cuanto a la bibliografía sobre eventos sociales y culturales, la misma es escasa. Se encuentra el problema de avisos publicitarios de TV, [9], en donde se propone cambiar la oferta de un aviso de 30 segundos a paquetes de anuncios con diferentes características, (precio, cantidad de avisos que contiene el paquete, horario del día en el cual será publicitado). En este caso, se estudian diversas resoluciones mediante enfoques que incluyen Programación Lineal, Relajación Lagrangiana, Programación con Restricciones y Búsqueda Local. En otro ámbito, también se encuentra el problema de un encuentro de yates, en donde se designan yates “anfitriones” y yates “invitados” [10]. El evento consistía en que la tripulación de los yates invitados visitaría los yates anfitriones en seis períodos sucesivos de media hora. El objetivo del problema es minimizar los yates anfitriones dado que éstos debían ser abastecidos con comida y debían contar con otros requisitos previos, lo que significaba un gasto. Los autores aplican ambos enfoques, Programación Lineal y Programación con Restricciones, resultando que mientras que utilizando la primera no se logra encontrar una solución, con la segunda sí.

Además de las categorías mencionadas previamente, no se relevan artículos relacionados con programación de eventos. Es importante destacar esto ya que, si bien tanto la Programación con Restricciones como la Programación Matemática han sido, (y continúan siendo), muy utilizadas en la resolución de problemas de asignación, problemas de transporte y programación de calendarios, (principalmente deportivos), se encuentra que es una herramienta con la cual aún no se ha experimentado demasiado en el ámbito de la programación de eventos sociales y/o culturales.

En este punto es importante mencionar el trabajo que antecede al presentado en este documento, realizado por A. Cardozo, J. C. Machin y C. Guido [11]. El mismo tuvo como uno de sus objetivos confeccionar el fixture del Carnaval de las Promesas, edición 2020, mediante métodos cuantitativos. Dentro del ámbito artístico, en este trabajo se presenta el relevamiento de [12], [13] y [14], tres trabajos que se basan en el problema mencionado anteriormente, de comerciales de TV. En [12] el problema consiste en que luego de aceptar los anuncios, se haga la programación de los mismos con el objetivo de maximizar ganancias. En [13], posteriormente al análisis sobre si aceptar o no una solicitud de anuncio, se programan los anuncios aceptados. En estos dos últimos casos se utiliza métodos cuantitativos. Finalmente, en [14], mediante un algoritmo heurístico se genera un plan de ventas óptimo que satisfaga los requisitos de los anunciantes.

2.2.1. Programación con Restricciones

La Programación con Restricciones es un paradigma de la programación en informática donde las relaciones entre las variables son expresadas mediante restricciones. Es una técnica utilizada para describir y resolver muchos problemas de la vida real como por ejemplo problemas de planificación y toma de decisiones entre otros [15]. Consiste en modelar el problema mediante la declaración de restricciones y encontrar soluciones que satisfagan las mismas. Este tipo de problemas de satisfacción con restricciones son conocidos como CSP, (Constraint Satisfaction Problem).

La resolución de problemas de CSP se puede considerar en dos etapas:

La primera consiste en modelar el problema. Este está determinado por un conjunto de variables, cada una asociada a un dominio y un conjunto de restricciones sobre esas variables.

Luego, en una segunda etapa queda determinada la solución mediante la satisfacción de las restricciones. Es en esta fase es que cobran importancia el uso de técnicas para la eliminación de valores inconsistentes en los dominios de las variables, así como también la utilización de algoritmos de búsqueda. En general, éstas se combinan para realizar una búsqueda más eficiente.

2.2.1.1. Conceptos

Un problema de Programación con Restricciones quedará determinado mediante [25]:

- Un conjunto de variables X $\{x_1;....; x_n\}$: estas variables quedan definidas por las características del problema y son las incógnitas que se intentarán resolver.
- $D = < D_1;....; D_n >$: donde cada elemento representa el dominio que contiene los valores posibles que se le pueden asignar a cada variable.
- R es el conjunto que define las restricciones: éstas restringen los valores que las variables pueden tomar. En el caso de las restricciones, existen tres tipos: unarias, binarias y no-binarias, considerando una, dos o muchas variables asociadas respectivamente.
  Ejemplos:
  - La restricción $x < 18$ es una restricción unaria sobre la variable x.
  - La restricción $x_1 + x_2 = 4$ es una restricción binaria.

Si éste además es un problema de optimización de restricciones, la solución estará asociada a una función objetivo:

- Una función objetivo es una expresión que relaciona variables del problema con el objetivo de obtener un valor que interesa optimizar.

La asignación de variables implica almacenar un valor en el espacio de memoria correspondiente a cada una de ellas. Cuando todas las variables son asignadas y estos valores satisfacen todas las restricciones, estas forman parte de una solución global. En caso de que esta asignación sea parcial, es decir, que se asignen valores a un conjunto de variables del problema y estas satisfagan todas las restricciones, se dice que la solución es parcial.

A continuación, se detallan las principales técnicas para procesar las restricciones.

- Algoritmos de búsqueda: éstos son un conjunto de instrucciones que exploran el espacio de soluciones con el objetivo de encontrar una solución cumpliendo con las condiciones del problema.
- Técnicas de consistencia: estas técnicas tienen como objetivo eliminar valores inconsistentes de los dominios de las variables para, de esta forma, reducir el espacio de búsqueda complementando a los algoritmos de búsqueda.

- Heurísticas de ordenación de variables: consiste en seleccionar el orden en el cual las variables van a ser estudiadas e instanciadas para, de esta manera, mejorar la eficiencia de resolución.

2.2.1.2. Algoritmos de Búsqueda

Los algoritmos de búsqueda son un conjunto de instrucciones que exploran el espacio de soluciones generadas por las posibles asignaciones de valores que pueden tomar las variables del problema con el objetivo de encontrar una solución en caso de que exista [16]. La combinación de estas asignaciones de valores genera un espacio de búsqueda denominado comúnmente árbol de búsqueda.

En la Figura 1 se puede observar un árbol de búsqueda generado por las posibles asignaciones.

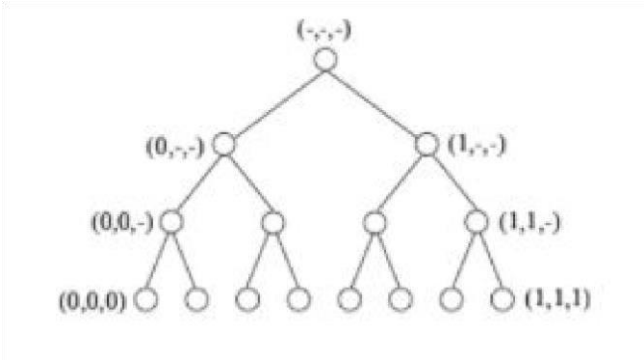

Figura 1: Árbol de Búsqueda [16].

Este árbol de búsqueda queda definido por nodos intermedios y nodos hojas. Los nodos intermedios representan las asignaciones parciales de variables, es decir que no todas las variables están asignadas y los nodos hojas, de menor nivel, representan las posibles asignaciones de la totalidad de las variables representadas.

Los algoritmos de búsqueda se pueden clasificar en algoritmos completos e incompletos [25]. Los algoritmos completos exploran todo el árbol de búsqueda obteniendo una solución óptima o demostrando que el problema no tiene solución. Los algoritmos incompletos son más eficientes debido a que realizan la búsqueda en un espacio de solución acotado, es decir, que no exploran todo el árbol de búsqueda, pero por este motivo no garantizan que la solución encontrada sea óptima.

Se presentan a continuación los siguientes algoritmos sistemáticos: Generación y prueba, Backtracking y Chequeo hacia adelante.

2.2.1.2.1. Generación y Prueba

Este algoritmo recorre el árbol de búsqueda generado por el espacio de soluciones realizando todas las combinaciones de asignaciones completas [17]. Luego verifica si estas cumplen con las restricciones del problema, como se puede observar en la Figura 2. Por este motivo resulta ineficiente ya que a medida que realiza asignaciones parciales no verifica si estas pueden formar parte de la solución.

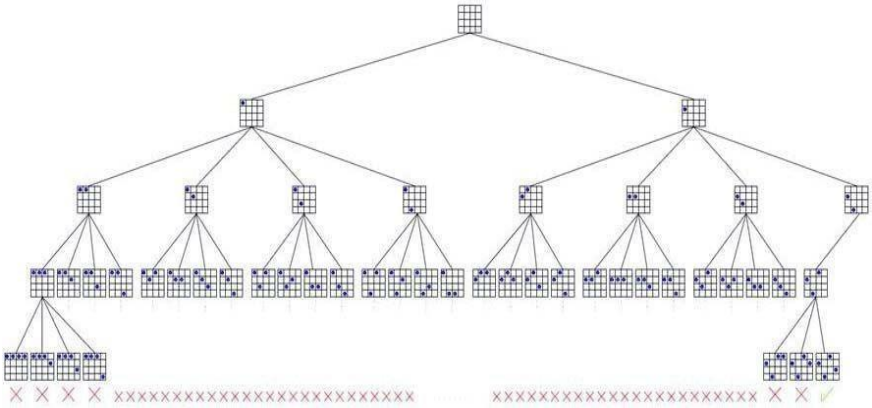

Figura 2: Ejemplo de búsqueda utilizando Generación y Prueba [17].

2.2.1.2.2. Backtracking

El algoritmo Backtracking explora el árbol de búsqueda en profundidad, pero a medida que asigna un valor a la variable actual, verifica que sea consistente con la asignación parcial [25].

Si el valor de la variable actual es consistente con la asignación parcial, el algoritmo avanza en el árbol de búsqueda seleccionando una nueva variable realizando la misma verificación.

Si el valor de la variable actual no es consistente con la asignación parcial, el algoritmo retrocede y asigna un nuevo valor a la variable actual. Se repite este procedimiento hasta encontrar que forme parte de una solución parcial, avanzando nuevamente en el árbol de búsqueda o retrocediendo en caso de que agote el dominio que puede tomar la misma, como se puede observar en la Figura 3.

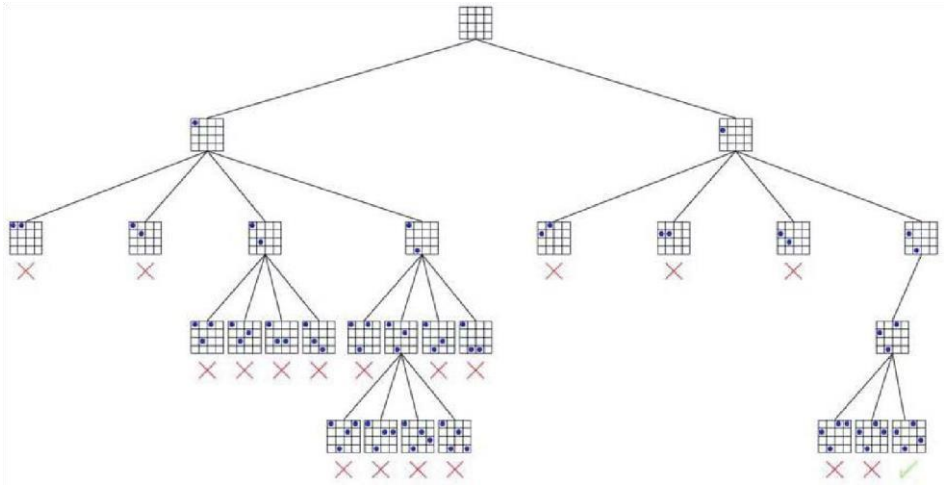

Figura 3: Ejemplo de búsqueda utilizando Backtracking [17].

El algoritmo finaliza cuando todas las variables son consistentes y satisfacen todas las restricciones o cuando todas las combinaciones de los valores que pueden tomar las variables no conducen a una solución.

Si bien este algoritmo es similar al algoritmo de Generación y Prueba, resulta más eficiente que éste porque no sigue explorando en una rama de asignaciones parciales que no conduce a una solución [16]. De todas formas, es considerado ineficiente porque chequea la variable actual y las pasadas sin considerar si la variable actual es consistente con las futuras.

2.2.1.2.3. Chequeo hacia adelante

El algoritmo Chequeo hacia adelante, más conocido como Forward Checking, a diferencia de los algoritmos de Backtracking y Generación y prueba, realiza una verificación hacia adelante en cada una de las etapas de la búsqueda utilizando técnicas de consistencia.

De esta forma, verifica que la asignación actual sea consistente con los valores de las variables futuras, eliminando temporalmente los valores inconsistentes de los dominios de las variables futuras [16].

En el caso de que todos los posibles valores del dominio de una variable futura sean inconsistentes con la variable actual, se asigna un nuevo valor del dominio a la variable actual. Así es que, si ningún valor es consistente se realizará un backtrack explorando una nueva variable, como se observa en la Figura 4.

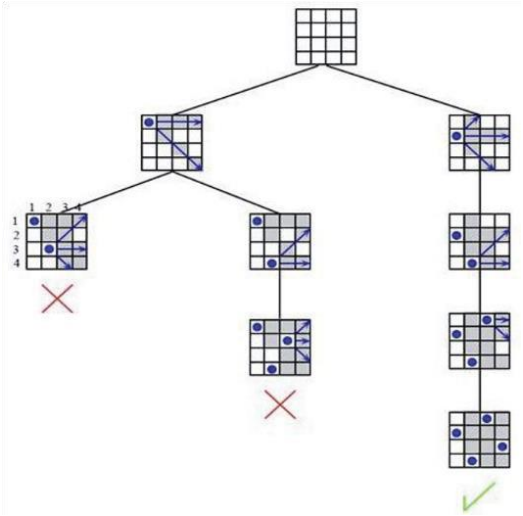

Figura 4: Ejemplo de búsqueda Chequeo hacia Adelante [16].

2.2.1.3. Técnicas de Consistencia

Las técnicas de consistencia son utilizadas para detectar aquellos valores de los dominios de las variables que son inconsistentes para, de esta forma, reducir el espacio de búsqueda complementándose con los algoritmos de búsqueda logrando mayor eficiencia en términos de tiempos de resolución.

2.2.1.3.1. Consistencia de Nodo

Este nivel de consistencia verifica todas las restricciones unarias sobre una variable eliminando del dominio de esta variable los valores que no cumplen con estas restricciones. De esta forma se logra reducir el espacio de búsqueda [16].

Ejemplo: La variable x, cuyo dominio es [0;110], y las restricciones unarias $x \geq 8$, $x \leq 99$. Los intervalos [0;7] y [100;110] son eliminados debido a que estos no conducirán a ninguna solución.

2.2.1.3.2. Consistencia de Arco

Este nivel de consistencia considera las restricciones binarias. Es decir, que dadas dos variables x_i y x_j, para cada valor perteneciente al dominio de x_i verifica si existe algún valor del dominio de x_j que satisface las restricciones que relacionan x_i y x_j. De esta forma, los valores del dominio de x_i que no son consistentes con ningún valor del dominio de la variable x_j, son eliminados [17].

Ejemplo. Dadas dos variables x_i de dominio [2;5], x_j de dominio [9;11] y la restricción binaria $x_i < x_j$ se puede observar que todos los valores del dominio de x_i son consistentes.

2.2.1.4. Heurísticas de ordenación

De forma de lograr que los algoritmos de búsqueda sean más eficientes en términos de tiempo de resolución, es importante definir estrategias de enumeración seleccionando el orden en el cual las variables van a ser estudiadas e instanciadas [16].

Esto es posible lograrlo mediante la utilización de heurísticas para la ordenación de variables y valores.

Las heurísticas de ordenación de variables tienen como objetivo definir el orden en el que las variables van a ser asignadas. Se priorizan las variables más restringidas para, de esta forma, identificar y eliminar del árbol de búsqueda aquellas asignaciones que no conducen a una solución.

Las heurísticas para la ordenación de valores tienen como objetivo seleccionar el valor de la variable actual que es más probable que conduzca a una solución. Para ello se priorizan en la búsqueda los valores de la variable actual que generen menos inconsistencias a las asignaciones futuras.

2.3. Carnaval uruguayo y Carnaval de las Promesas

El Carnaval uruguayo se festeja desde tiempos de la Colonia [18]. Fue en esta época donde algunas prácticas y tradiciones europeas asociadas a diferentes leyendas y festejos religiosos sirvieron para dar indicios del Carnaval [19]. Esta celebración sirvió como una forma de expresión a los nuevos pobladores, que en sus comienzos se asemejaba mucho a los festejos de la antigua Roma, asociados con la catarsis y la exaltación de la carne. Durante esos años los desfiles de Carnaval abarcaban un largo recorrido: partían de la Plaza Constitución, recorrían varias calles de la Ciudad Vieja, pasaban por 18 de Julio, llegaban hasta la Plaza Cagancha y regresaban al punto de partida por otro recorrido. Los avances tecnológicos trajeron la iluminación, primero a gas y después a electricidad, que engalanaron las avenidas cada año con más esplendor.

Se registran en nuestro país los primeros festejos populares del Carnaval sobre el año 1750: donde la gente se tiraba huevos, harinas y diferentes productos como parte de la celebración. Años después se intentó organizar este festejo poniéndole fin a las manifestaciones de este tipo, priorizando las expresiones artísticas, como el canto y la actuación. De esta forma comenzó a institucionalizarse el Carnaval como una manifestación cultural que sirvió además para que los ciudadanos de la naciente nación pudieran sociabilizar y sentirse identificados con una expresión popular nacional, creadora de identidad.

Entre finales del siglo XIX y comienzos del XX, nuestro Carnaval se convirtió en uno de los más largos del mundo con una duración de cuarenta días. Durante todo febrero y parte de marzo las agrupaciones actúan en escenarios llamados tablados, y en el concurso oficial en el Teatro de Verano Ramón Collazo, en Montevideo, aunque también es importante destacar que en muchas ciudades del interior también se festeja el Carnaval. Las categorías que participan de esta festividad son Murgas, Revistas, Parodistas, Humoristas y Lubolos.

Se muestran imágenes del Carnaval uruguayo en las Figuras 5 a 8 respectivamente.

Figura 5: Carnaval uruguayo [18].

Figura 6: Murga Agarrate Catalina [20].

Figura 7: Desfile Inaugural edición 2020 [21].

Figura 8: Primera etapa de la rueda 1, Carnaval 2020 [20].

El Carnaval de las Promesas nace en el año 1988, organizado por el Departamento de Cultura de la Intendencia de Montevideo, a través de la Gerencia de Festejos y Espectáculos, en conjunto con ADICAPRO, (Asociación de Directores del Carnaval de las Promesas), [22]. ADICAPRO es una organización sin fines de lucro que tiene como objetivo el desarrollo artístico, personal y cultural de niños y adolescentes participantes del denominado Carnaval de las Promesas, de diversas zonas de Montevideo, e incluso, del interior del país. Se muestra en la Figura 9, niños participantes del Carnaval de las Promesas.

Figura 9: Carnaval de las promesas [22].

El objetivo de este evento es promover la inclusión social y la integración de niñas, niños y adolescentes de todo el territorio de Montevideo y área metropolitana. En el caso del objetivo de la integración social, el mismo se busca plasmar a través del carnaval y su diversidad de expresiones artísticas, como excelente oportunidad para la generación de valores e identidad cultural. También se buscará fomentar y promover políticas específicas destinadas a infancia y juventud, como segmentos etarios principales para las políticas culturales del Departamento de Cultura. Dichas políticas contemplarán la formación en valores democráticos de participación, diversidad, tolerancia y perspectiva de género.

El Carnaval de las Promesas se lleva a cabo desde diciembre a enero en el Teatro de Verano, presentándose más de 40 agrupaciones integradas por jóvenes de hasta 18 años de edad en todas las categorías carnavalescas: sociedad de negros y lubolos, revistas, murgas, parodistas y humoristas. Está constituido por los eventos: Encuentro Evaluatorio, Desfile Oficial y Concurso Oficial de Agrupaciones Infantiles, a desarrollarse en el Teatro de Verano Ramón Collazo.

Según el Reglamento del Concurso Oficial de Agrupaciones Infantiles “Carnaval de las Promesas”, cada agrupación deberá contar con un/a Director/a, persona/s con dieciocho (18) años cumplidos, que asumen la responsabilidad de cada agrupación, a los efectos de su presentación en el Concurso Oficial de Agrupaciones Infantiles.

El reglamento consiste en 11 capítulos que determinan el correcto desarrollo del concurso [23]. Se listan a continuación los capítulos del reglamento.

- Capítulo I: Organización. Misión y Visión. Eventos que constituyen el Carnaval de las Promesas.
- Capítulo II: Directores/as responsables.
- Capítulo III: Denominaciones de las agrupaciones.
- Capítulo IV: Inscripción al Concurso.
- Capítulo V: Encuentro evaluatorio.
- Capítulo VI: Categorías.
- Capítulo VII: Jurado del Concurso.
- Capítulo VIII: Definición de los Rubros
- Capítulo IX: De las actuaciones en el Concurso Oficial de Agrupaciones Infantiles
- Capítulo X: Las Sanciones
- Capítulo XI: Disposiciones Generales

3. Caso de estudio

En esta sección se expone la descripción del problema. Se comienza con una breve introducción sobre la estructura del Carnaval, (rondas, etapas, horarios, agrupaciones), luego se plantea el problema con la formulación matemática correspondiente y finalmente se presenta el modelado con Programación Lineal Entera y Programación con Restricciones: los calendarios resultantes, los tiempos de ejecución y la comparación de los resultados obtenidos.

En esta etapa fue de vital importancia el vínculo con la Gerencia de Eventos de la Intendencia de Montevideo y la Asociación de Directores de Carnaval (ADICAPRO), esto permitió comprender la problemática y las características propias de la competencia e interiorizarse en el reglamento y las particularidades de la competencia para la edición 2021.

3.1. Introducción

El concurso consta de dos rondas o ruedas; cada ronda se compone de 10 etapas, (corresponden a 10 noches diferentes), en las que actúan la totalidad de las agrupaciones una única vez por ronda. Cada etapa se compone originalmente de 4 horarios: cada uno correspondiente a la actuación de una agrupación.

- Etapa: Es un día de actuación. Normalmente actúan en el entorno de 4 agrupaciones, dependiendo de la cantidad de agrupaciones totales.
- Horario: Turno de una agrupación dentro de una etapa.

Así es que, en el caso de que el concurso esté conformado por 40 agrupaciones, el mismo se compone de 2 rondas de 10 etapas cada una, en el que actúan 4 grupos en cada horario, (primer, segundo, tercer y cuarto horario).

La organización de la edición del Carnaval de las Promesas 2021 se llevó a cabo bajo un marco de incertidumbre debido al arribo de la pandemia aproximadamente en marzo de 2020, esto implicó reformular el formato habitual, teniendo en cuenta las disposiciones del gobierno, los protocolos referidos a espectáculos públicos, aforos, locación y otras consideraciones que surgían con el avance de la misma.

A mediados de noviembre del 2020 se preveía una modalidad que, si bien mantenía la actuación de 4 agrupaciones por etapa, entre las actuaciones del segundo y tercer horario se realizaría un “entretiempo” que permitiera sanitizar los espacios de concurrencia.

Esta nueva modalidad fue comunicada a los autores de este proyecto en una reunión llevada a cabo en la Intendencia de Montevideo. Además, también se comunicó que el inicio del Carnaval de las Promesas sería el 19 de diciembre, y el final el 17 de enero: estas fechas abarcaban 2 rondas de 10 etapas cada una, en las cuales actuaban 4 agrupaciones, más una noche de premiación. En esta edición participarían 40 agrupaciones, conformadas por 1250 niños y jóvenes. Las etapas se dividirían en dos bloques: en vez de actuar 4 agrupaciones sin interrupciones, como era usual hasta el año 2020, actuarían dos agrupaciones seguidas de un entretiempo para sanitizar el Teatro de Verano, (lugar donde se iban a desarrollar las distintas etapas), y luego las dos restantes, manteniendo así la actuación de 4 agrupaciones por etapa.

Las entradas iban a ser ahora para cada bloque, (2 agrupaciones), y no para la etapa entera, (4 agrupaciones). Por esto es que esta nueva modalidad se podía entender como dos etapas en

una: de esta forma iban a “cerrar” dos agrupaciones y no una como hasta la última edición. Esto implicaba un nuevo criterio a la hora de establecer restricciones en el modelo matemático.

ADICAPRO definió cuáles agrupaciones podrían actuar en cada horario y, además, que las agrupaciones que actuaran en el primer horario de la primera ronda, debían actuar en el tercer horario de la segunda ronda y viceversa. En el caso de las agrupaciones que actuaban en el segundo y cuarto horario en la primera ronda, mantendrían estos horarios en la segunda ronda. Estas agrupaciones debían tener mayor convocatoria que las de los horarios primero y tercero dado que “cerraban” cada bloque. Esto se entiende de la siguiente manera: si las agrupaciones que tienen mayor convocatoria actuaran en los primeros horarios, las agrupaciones que cerrarían los bloques no tendrían demasiado público. Al asignarlas en los últimos horarios de los bloques, el público se quedaría hasta el final, presenciando ambos grupos. Lo usual es que los grupos que cierran pertenezcan a las categorías revistas, parodistas y murgas, siendo éstas las de mayor convocatoria.

Se listan en la siguiente tabla, (Tabla 1), las agrupaciones participantes con los horarios asignados por ADICAPRO.

Tabla 1: Agrupaciones participantes.

HORARIO	NRO.	AGRUPACIÓN	HORARIO	NRO.	AGRUPACIÓN
PRIMERA	29	LOS TOBY'S	TERCERA	23	CACHIRULOS
	19	FRENESI		28	LOS CHAPITAS
	22	COLIBRIQUIS		21	GNOMOS
	24	HEREDEROS		20	VALU'S
	25	DEPORTADOS		26	BAM BAM
	27	YOGUIS		15	CELESTINOS
	30	DESCARA2		16	PRINCIPES
	11	ZERO		17	TOON'S
	12	CHERRY'S		18	ZABRITOS
	39	S.C.		13	TROYANOS
SEGUNDA	6	FENIX	CUARTA	10	BUBY'S BIS
	4	ZODIACO		9	QUIJOTES
	8	MANDALA		14	WONKA'S
	7	JADE		3	DULCINEA
	36	DIABLITOS VERDES		1	SAPHIRUS
	35	LA DESCOCADA		2	ADRENALINA
	33	LOS DUENDES		5	LOS BUSCADORES
	37	DON BARULLO		32	LA ZAFADA
	40	OHANA		34	LOS PEPINITOS
	38	COSME Y DAMIAN		31	MANO A MANO

HUMORISTAS
PARODISTAS
LUBOLOS
REVISTAS
MURGA

El objetivo del modelo es balancear la concurrencia del público en las distintas etapas. Sin embargo, dada la situación particular de esta edición, lo que se necesitaría balancear sería la concurrencia de los bloques y no de las etapas. La metodología que se utilizó para balancear la concurrencia se explica a continuación.

Para poder cuantificar la convocatoria de una agrupación, se le asignó un valor a cada una, (ponderación), en el rango del 1 al 10: valores más altos se traducen en mayor convocatoria. Si bien en la edición 2019 del concurso este valor asignado surgió de una votación anónima entre las agrupaciones participantes, dada la incertidumbre en cuanto a la realización del Carnaval, no

fue hasta mediados de noviembre que ADICAPRO se comunicó con la Facultad de Ingeniería para dar aviso de la necesidad de confeccionar un fixture previo al día 21 del mes de diciembre. La celeridad con la que se necesitaba el calendario no permitió que se realizara una nueva votación, por lo que ADICAPRO fue quien asignó dichos valores, basándose en la encuesta realizada para la edición anterior, en el trabajo de A. Cardozo, J.C. Machin y C. Guido [11]. Se listan los resultados en Anexo 2.

Debido al corto plazo del que se dispuso, los diferentes calendarios que se presentaron a ADICAPRO se realizaron mediante Programación Lineal Entera y no con Programación con Restricciones, como era la idea inicial. A pesar de esto, en los meses consecutivos sí se realizaron nuevos modelos con esta metodología, los cuales se tratarán más adelante en este documento.

3.2. Formulación matemática

Para la formulación matemática se revisó detalladamente la realizada en [11], realizando las modificaciones correspondientes.

Primero se definen los conjuntos que se van a utilizar: conjunto de etapas, (DN), conjunto de horarios, (T) y conjunto de rondas, (C). Se definen los conjuntos correspondientes a las categorías, (murgas, parodistas, humoristas, lubolos y revistas y el conjunto de agrupaciones que es la unión de los anteriores, (A). Luego, se definen los conjuntos de las agrupaciones habilitadas a actuar según horario: el conjunto $A1$ especifica los conjuntos que pueden actuar en el primer horario, $A2$ define los conjuntos que pueden actuar en el segundo horario, y de la misma manera $A3$ y $A4$ definen lo análogo con los horarios tercero y cuarto. Vale aclarar que la definición de estos últimos conjuntos es propia de la formulación matemática para la edición 2021 del concurso. Los mismos son definidos por ADICAPRO debido a las restricciones sanitarias que podrían llegar a existir. Por último, se define el conjunto K que especifica los tipos de categoría a los cuales puede pertenecer una agrupación: 1 corresponde a murgas, 2 a lubolos, 3 a humoristas, 4 a parodistas y 5 a revistas.

Posteriormente a definir los conjuntos se definen los parámetros: V_i corresponde a la ponderación de cada agrupación i, la misma es obtenida como se explicó anteriormente. El parámetro w corresponde al promedio de ponderaciones, es decir, el promedio de todos los valores V_i. El parámetro F_i define a qué categoría pertenece la agrupación i.

Finalmente, se definen las variables. La variable de decisión es $x_{i,d,t,c}$, la cual vale 1 si la agrupación i actúa en el horario t de la etapa d, en la ronda c. Las variables auxiliares son z_c e $y_{d,c}$: z_c es la variable para medir la desviación de la ponderación promedio en cada ronda, $y_{d,c}$ es la suma de las ponderaciones de las agrupaciones que actúan en la etapa d de la ronda c.

Después de las definiciones anteriores se realiza la formulación matemática. Vale aclarar que los cambios con respecto a la formulación matemática realizada en 2020 son los que se mencionan a continuación: se agregan las restricciones de (7) a (12), (son restricciones en base a la definición de conjuntos habilitados a actuar según horario, lo cual no se realizó hasta 2020) y se eliminan restricciones de la formulación anterior que no aplicaban en esta edición del evento. Entre ellas se encuentran, por ejemplo, las restricciones relativas a la actuación de bandas, (en la edición 2020 no había agrupaciones que contaran con banda), así como restricciones sobre los horarios predominantes según categoría de las agrupaciones.

Se presenta a continuación el modelo algebraico del problema.

CONJUNTOS

D: Conjunto de etapas, con cardinalidad de $D=DN$.

T: Conjunto de horarios, con cardinalidad de $T=TN$.

C: Conjunto de rondas, con cardinalidad de $C=CN$.

M: Conjunto de murgas, con cardinalidad de $M=MN$.

P: Conjunto de parodistas, con cardinalidad de $P=PN$.

H: Conjunto de humoristas, con cardinalidad de $H=HN$.

L: Conjunto de lubolos, con cardinalidad de $L=LN$.

R: Conjunto de revistas, con cardinalidad de $R=RN$.

$A = M \cup P \cup H \cup L \cup R \cup I$: Conjunto de agrupaciones, con cardinalidad de $A = AN$.

$A1$: Conjunto de agrupaciones que pueden actuar en el primer horario en la ronda 1, con cardinalidad de $A1=A1N$.

$A2$: Conjunto de agrupaciones que pueden actuar en el segundo horario en la ronda 1, con cardinalidad de $A2=A2N$.

$A3$: Conjunto de agrupaciones que pueden actuar en el tercer horario en la ronda 1, con cardinalidad de $A3=A3N$.

$A4$: Conjunto de agrupaciones que pueden actuar en el cuarto horario en la ronda 1, con cardinalidad de $A4=A4N$.

K = {1 . . 5} : Conjunto de tipo de categoría a la que pertenece la agrupación, con: 1= murga, 2= lubolos, 3= humoristas, 4= parodistas y 5= revistas. Con cardinalidad de $K=KN$.

PARÁMETROS

V_i: Ponderación para cada agrupación i, indica la popularidad en términos de convocatoria.

$w = \frac{\sum_{i \in A} v_i}{AN}$: Promedio de ponderaciones

$F_i \in K$: Parámetro para agrupación i que define el tipo de categoría a la que pertenece la agrupación.

VARIABLES

Variables de decisión

$x_{i,d,t,c} = \begin{cases} 1 & \text{si la agrupación } i \text{ actúa en la etapa } d, \text{ en el horario } t, \text{ en la ronda } c. \\ 0 & \text{si no.} \end{cases}$

Variables auxiliares

z_c : variable auxiliar para medir la desviación de la ponderación promedio en cada ronda *c*.

$y_{d,c}$: $d \in D$, $c \in C$ Ponderación para la etapa d, en la ronda c. Es la suma de las ponderaciones de las agrupaciones que actúan esa etapa.

FORMULACIÓN MATEMÁTICA

$$\min \sum_{c \in C} z_c \quad (1)$$

s.a:

$$y_{d,c} = \frac{\sum_{i \in A} \sum_{t \in T} v_i x_{i,d,t,c}}{4} \quad \forall\, d \in D, \forall\, c \in C \quad (2)$$

$$z_c \geq y_{d,c} - w \quad \forall\, d \in D, \forall\, c \in C \quad (3)$$

$$z_c \geq w - y_{d,c} \quad \forall\, d \in D, \forall\, c \in C \quad (4)$$

$$\sum_{t \in T} \sum_{d \in D} x_{i,d,t,c} = 1 \quad \forall\, i \in A, \forall\, c \in C \quad (5)$$

$$\sum_{i \in A} x_{i,d,t,c} = 1 \quad \forall\, t \in T, \forall\, d \in D, \forall\, c \in C \quad (6)$$

$$x_{i,d,4,c} = 0 \quad \forall\, i \in A \backslash A4, \forall\, d \in D, \forall\, c \in C \quad (7)$$

$$x_{i,d,1,1} = 0 \quad \forall\, i \in A \backslash A1, \forall\, d \in D \quad (8)$$

$$x_{i,d,2,c} = 0 \quad \forall\, i \in A \backslash A2, \forall\, d \in D, \ \forall\, c \in C \quad (9)$$

$$x_{i,d,3,1} = 0 \quad \forall\, i \in A\backslash A3, \forall\, d \in D \tag{10}$$

$$x_{i,d,3,2} = 0 \quad \forall\, i \in A\backslash A1, \forall\, d \in D \tag{11}$$

$$x_{i,d,1,2} = 0 \quad \forall\, i \in A\backslash A3, \forall\, d \in D \tag{12}$$

$$x_{i,d,t,c} + x_{j,d,t+1,c} \leq 1 \quad \forall\, d \in D,\ \forall\, c \in C,\ \forall\, t \in \{1,3\},\ \forall\, i,j \in A,\ F_i = F_j \tag{13}$$

$$\sum_{i \in L} \sum_{t \in T} x_{i,d,t,c} \leq 1 \quad \forall\, d \in D,\ \forall\, c \in C \tag{14}$$

$$\sum_{i \in M} \sum_{t \in T} x_{i,d,t,c} \leq 1 \quad \forall\, d \in D,\ \forall\, c \in C \tag{15}$$

$$\sum_{i \in R} \sum_{t \in T} x_{i,d,t,c} \leq 1 \quad \forall\, d \in D,\ \forall\, c \in C \tag{16}$$

$$\sum_{i \in P} \sum_{t \in T} x_{i,d,t,c} \leq 1 \quad \forall\, d \in D,\ \forall\, c \in C \tag{17}$$

$$\sum_{i \in H} \sum_{t \in T} x_{i,d,t,c} \geq 1 \quad \forall\, d \in D,\ \forall\, c \in C \tag{18}$$

$$\sum_{i \in H} \sum_{t \in T} x_{i,d,t,c} \leq 2 \quad \forall\, d \in D,\ \forall\, c \in C \tag{19}$$

$$\sum_{i \in P} x_{i,d,4,c} + \sum_{i \in P} x_{i,d+1,4,c} = 1 \quad \forall\, d \in [1,(DN-1)],\ \forall\, c \in C \tag{20}$$

$$\sum_{i \in R} x_{i,d,4,c} + \sum_{i \in R} x_{i,d+1,4,c} = 1 \quad \forall\, d \in [1,(DN-1)],\ \forall\, c \in C \tag{21}$$

$$\sum_{i \in M} x_{i,d,4,c} + \sum_{i \in M} x_{i,d+1,4,c} = 1 \quad \forall\, d \in [1,(DN-1)],\ \forall\, c \in C \tag{22}$$

$$\sum_{t \in T} x_{i,1,t,1} \leq \sum_{d \in [1,4]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{23}$$

$$\sum_{t \in T} x_{i,2,t,1} \leq \sum_{d \in [1,4]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{24}$$

$$\sum_{t \in T} x_{i,3,t,1} \leq \sum_{d \in [1,5]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{25}$$

$$\sum_{t \in T} x_{i,4,t,1} \leq \sum_{d \in [1,6]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{26}$$

$$\sum_{t \in T} x_{i,5,t,1} \leq \sum_{d \in [3,7]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{27}$$

$$\sum_{t \in T} x_{i,6,t,1} \leq \sum_{d \in [4,8]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{28}$$

$$\sum_{t \in T} x_{i,7,t,1} \leq \sum_{d \in [5,10]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{29}$$

$$\sum_{t \in T} x_{i,8,t,1} \leq \sum_{d \in [6,10]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{30}$$

$$\sum_{t \in T} x_{i,9,t,1} \leq \sum_{d \in [7,10]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{31}$$

$$\sum_{t \in T} x_{i,10,t,1} \leq \sum_{d \in [7,10]} \sum_{t \in T} x_{i,d,t,2} \quad \forall\, i \in A \tag{32}$$

$$x_{i,d,t,c} \in \{0,1\} \quad \forall\, i \in A,\ \forall\, d \in D,\ \forall\, t \in T,\ \forall\, c \in C \tag{33}$$

La función objetivo, (1), minimiza la sumatoria de las desviaciones de la ponderación promedio de cada ronda, con el fin de balancear la convocatoria en los diferentes bloques de cada etapa. La ecuación (2) calcula y; es la ponderación de la convocatoria de la etapa, dividido 4 que es la cantidad de horarios de una etapa. En las familias de ecuaciones (3) y (4) se define la variable auxiliar z_c, como la diferencia entre el promedio de ponderaciones w y la ponderación por etapa $y_{d,c}$.

La familia de restricciones (5) garantiza que cada agrupación actúa una sola vez por ronda, mientras que (6) define que, a cada horario disponible, (en todos los horarios de cada etapa, de cada ronda), le corresponde una agrupación.

Las familias de ecuaciones (7) y (9) aseguran que tanto en la ronda 1 como en la 2, actúen las agrupaciones habilitadas para los horarios segundo y cuarto. Cabe recordar que las agrupaciones habilitadas a actuar en estos horarios son las mismas para ambas rondas ya que a diferencia de los grupos que actúan a primera y tercera hora, estos sí son estáticos en ambas rondas, manteniendo su horario. Las familias de ecuaciones (8) y (10) aseguran entonces que en el tercer horario de la primera ronda, (a diferencia de las restricciones mencionadas

anteriormente), actúen las agrupaciones habilitadas a actuar en esta hora, y lo equivalente para el primer horario; de la misma forma las familias de ecuaciones (11) y (12) garantizan que las agrupaciones que actúan en el primer horario y en el tercer horario en la primera ronda, actúen en el tercer horario y primer horario respectivamente en la segunda ronda.

La familia de ecuaciones (13) asegura que nunca actúen dos agrupaciones de la misma categoría en horarios consecutivos de un mismo bloque.

Las familias de restricciones de (14) a (17) garantizan que en todas las etapas actúen como máximo una agrupación de cada categoría, para las categorías que tienen menos cantidad de agrupaciones que etapas por ronda, (10 etapas): las mismas son lubolos, murgas, revistas y parodistas. Humoristas es la única categoría que tiene más de 10 agrupaciones: se compone de 12 agrupaciones. Esto significa que inevitablemente van a existir dos etapas en que van a actuar más de una agrupación de esta categoría. La familia de restricciones (18) asegura que en cada etapa va a actuar al menos una agrupación de humoristas, mientras que la familia de restricciones (19) asegura que en ninguna etapa van a actuar más de 2 de las mismas.

Como se explicó anteriormente, los cierres de las etapas son los horarios más codiciados por las agrupaciones participantes dado el prestigio que esto representa en la cultura carnavalesca: las familias de restricciones de (20) a (22) aseguran la alternancia de categorías en el cierre de las etapas, esto es que, si una agrupación actúa en el cuarto horario en la etapa d, no puede actuar otra agrupación de la misma categoría en la etapa $d + 1$. Se enuncian tres restricciones dado que son tres las categorías de las agrupaciones habilitadas para cerrar, (parodistas, revistas y murgas).

Existen también restricciones para el espacio de tiempo que debe haber entre la actuación de una agrupación en la primera ronda y en la segunda: no sería justo que una agrupación actúe con una diferencia de tres días, por ejemplo, entre la primera ronda y la segunda, mientras que otra agrupación actúe con el doble de tiempo o más. Por esto es que se definen las familias de restricciones desde la (23) a la (32). Las mismas se basan en la Tabla 2, el cual establece los posibles períodos de tiempo entre las actuaciones de una misma agrupación entre las dos rondas.

Tabla 2: Etapas habilitadas para actuar en la segunda ronda, según actuación en la primera ronda.

RONDA 1 \ RONDA 2	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x						
2	x	x	x	x						
3	x	x	x	x	x					
4	x	x	x	x	x	x				
5			x	x	x	x	x			
6				x	x	x	x	x		
7					x	x	x	x	x	x
8						x	x	x	x	x
9							x	x	x	x
10							x	x	x	x

Cabe aclarar que esta determinación es la misma que se utilizó en la última edición siendo que se consideró acertada.

Finalmente, la restricción (33) determina el dominio de la variable de decisión $x_{i,d,t,c}$.
Así es que el modelo resultante está constituido por:

- $(AN * DN * TN * CN)$ variables binarias $x_{i,d,t,c}$.
- $(DN * CN) + (CN)$ variables continuas $y_{d,c}$ y z_c respectivamente.
- $9 * (DN * CN) + (AN * CN) + (TN * DN * CN) + (A4N * DN * CN) + 2 * (A1N * DN) + (A2N * DN * CN) + 2 * (A3N * DN) + (DN * CN * 2 * 2 * AN * AN) + 3 * ((DN - 1) * CN) + 10 * (AN) + (AN * DN * TN * CN)$ restricciones.

Donde para la edición 2020/2021 $AN = 40, A1N = 10, A2N = 10, A3N = 10, A4N = 10, DN = 10, TN = 4, CN = 2, HN = 12, PN = 10, RN = 8, MN = 7, LN = 3$.
Contabilizando 3200 variables binarias, 22 variables continuas y 12914 restricciones.

3.3. Implementación con Programación Lineal Entera

Dada la celeridad con la que se necesitaba determinar el fixture para la edición 2020 del Carnaval de las Promesas, se utilizó el modelo de Programación Lineal Entera. Es importante mencionar en este punto que debido a un error de fórmula los fixtures que se presentan a continuación se obtuvieron ingresando un promedio de ponderaciones, (w), diferente al promedio real: el promedio de ponderaciones ingresado fue de 4,75 mientras que el promedio realmente correspondía a 5,41. Se presentan de igual manera estos calendarios debido a que fueron los que se presentaron a ADICAPRO para la elección del calendario definitivo. Sin embargo, los fixtures que se presentan en el capítulo de Experimentación numérica sí se obtuvieron en base al promedio real, dado que fue en este momento cuando se detectó el error.

El modelo se resolvió con el software AMPL/CPLEX 12.10.0.0 con un tiempo de ejecución límite de 3.600 segundos. Se utilizó una PC con las siguientes características: procesador Intel Core i7 - 5500U CPU, 2.40GHz y 8GB de memoria RAM. El fixture resultante se puede observar en la Figura 10. En lo que sigue, este modelo se denominará “Modelo R2”, (modelo de dos rondas).

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	YOGUIS	DEPORTADOS	FRENESI	ZERO	COLIBRIQUIS	HEREDEROS	DESCARA2	S.C.	LOS TOBY'S	CHERRY'S
2	MANDALA	LOS DUENDES	ZODIACO	OHANA	JADE	DIABLITOS VERDES	FENIX	LA DESCOCADA	COSME Y DAMIAN	DON BARULLO
3	BAM BAM	TOON'S	LOS CHAPITAS	CACHIRULOS	TROYANOS	CELESTINOS	ZABRITOS	GNOMOS	PRINCIPES	VALU'S
4	QUIJOTES	ADRENALINA	BUBY'S BIS	DULCINEA	LA ZAFADA	SAPHIRUS	LOS PEPINITOS	WONKA'S	MANO A MANO	LOS BUSCADORES
Prom. por etapa	5,44	4.85	5,42	5,43	5,38	5,43	6.54	6.08	4.71	5.05

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	LOS CHAPITAS	TOON'S	BAM BAM	CELESTINOS	TROYANOS	CACHIRULOS	VALU'S	PRINCIPES	GNOMOS	ZABRITOS
2	ZODIACO	LOS DUENDES	MANDALA	DIABLITOS VERDES	JADE	OHANA	DON BARULLO	COSME Y DAMIAN	LA DESCOCADA	FENIX
3	FRENESI	DEPORTADOS	YOGUIS	HEREDEROS	COLIBRIQUIS	ZERO	S.C.	DESCARA2	CHERRY'S	LOS TOBY'S
4	BUBY'S BIS	ADRENALINA	QUIJOTES	SAPHIRUS	LA ZAFADA	DULCINEA	WONKA'S	LOS PEPINITOS	LOS BUSCADORES	MANO A MANO
Prom. por etapa	5,42	5,44	5,44	5,43	5,38	5,43	5,42	5,40	5,36	5,42

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 10: Fixture resultante del modelo base de 2 rondas, (Modelo R2).

Con el objetivo de proporcionar a ADICAPRO más de un fixture, se agregaron diferentes restricciones que se describen a continuación. Éstas sirvieron para obtener variantes respecto al modelo base. De esta forma ADICAPRO podría elegir el que, según entendiera, mejor se ajustara al concurso.

En primer lugar, ADICAPRO sugirió una variante interesante. Esta implicaba que la última agrupación en actuar en la primera ronda dentro de las categorías Revistas y Murgas fueran las ganadoras en sus categorías en la edición anterior del concurso.

Para esto, se definió una nueva familia de variables, $b_{i,c}$:

$$b_{i,c} = \sum_{d \in D} \sum_{t \in T} [(10 \times d) + t] \times x_{i,d,t,c} \quad \forall\, i \in A, \forall\, c \in C \tag{34}$$

Esta definición implica que, dada una agrupación y una ronda determinadas, $b_{i,c}$ va a ser un número de 2 dígitos, donde el primer dígito indica la etapa en que actúa la agrupación y el segundo, el horario asignado.

A partir de esta definición, se formularon las siguientes restricciones:

$$b_{3,1} \geq b_{i,1} \quad \forall\, i \in R, i \neq 3 \tag{35}$$

$$b_{32,1} \geq b_{i,1} \quad \forall\, i \in M,\ i \neq 32 \tag{36}$$

A esta primera variante del modelo base se le denominó “Modelo 201”. Así es que se obtuvo el calendario que se muestra en la Figura 11.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	DEPORTADOS	DESCARA2	HEREDEROS	S.C.	COLIBRIQUIS	LOS TOBY'S	FRENESI	ZERO	YOGUIS	CHERRY'S
2	DON BARULLO	MANDALA	ZODIACO	LOS DUENDES	COSME Y DAMIAN	FENIX	DIABLITOS VERDES	JADE	LA DESCOCADA	OHANA
3	CELESTINOS	GNOMOS	TOON'S	LOS CHAPITAS	TROYANOS	VALU'S	PRINCIPES	CACHIRULOS	ZABRITOS	BAM BAM
4	SAPHIRUS	QUIJOTES	LOS PEPINITOS	BUBY'S BIS	LOS BUSCADORES	WONKA'S	ADRENALINA	MANO A MANO	DULCINEA	LA ZAFADA
Prom. por etapa	**5,46**	**5,47**	**5,47**	**5,38**	**5,39**	**5,39**	**6.54**	**6.08**	**4.71**	**5.05**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	TOON'S	LOS CHAPITAS	CELESTINOS	VALU'S	CACHIRULOS	TROYANOS	ZABRITOS	BAM BAM	PRINCIPES
2	MANDALA	ZODIACO	LOS DUENDES	DON BARULLO	FENIX	COSME Y DAMIAN	JADE	LA DESCOCADA	OHANA	DIABLITOS VERDES
3	DESCARA2	HEREDEROS	S.C.	DEPORTADOS	LOS TOBY'S	ZERO	COLIBRIQUIS	YOGUIS	CHERRY'S	FRENESI
4	QUIJOTES	LOS PEPINITOS	BUBY'S BIS	SAPHIRUS	WONKA'S	LOS BUSCADORES	MANO A MANO	DULCINEA	LA ZAFADA	ADRENALINA
Prom. por etapa	**5,47**	**5,47**	**5,38**	**5,46**	**5,39**	**5,29**	**5,42**	**5,40**	**5,43**	**5,44**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 11: Primer fixture alternativo, (Modelo 201).

Como segunda alternativa, se propuso algo similar a la primera: considerando las agrupaciones de la categoría Lubolos, se impone que una de ellas debería actuar en una etapa posterior a las que actuaran el resto de las agrupaciones de esta categoría en la primera ronda. De forma análoga se realizó la misma consideración para la categoría Parodistas. A este modelo se le llamó “Modelo 202”. Se utiliza en este caso también, la variable auxiliar definida anteriormente, $b_{i,c}$, y se obtiene el fixture que se muestra en la Figura 12.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	DESCARA2	YOGUIS	DEPORTADOS	LOS TOBY'S	S.C.	HEREDEROS	COLIBRIQUIS	CHERRY'S	ZERO	FRENESI
2	COSME Y DAMIAN	MANDALA	LOS DUENDES	FENIX	LA DESCOCADA	DIABLITOS VERDES	JADE	DON BARULLO	OHANA	ZODIACO
3	PRINCIPES	BAM BAM	TOON'S	ZABRITOS	GNOMOS	CELESTINOS	TROYANOS	VALU'S	COLIBRIQUIS	LOS CHAPITAS
4	DULCINEA	QUIJOTES	ADRENALINA	MANO A MANO	WONKA'S	SAPHIRUS	LA ZAFADA	LOS BUSCADORES	LOS PEPINITOS	BUBY'S BIS
Prom. por etapa	**5,43**	**5,44**	**5,44**	**5,42**	**5,36**	**5,43**	**5,38**	**5,42**	**6,01**	**5,42**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	BAM BAM	PRINCIPES	ZABRITOS	GNOMOS	TOON'S	TROYANOS	CELESTINOS	CACHIRULOS	VALU'S	LOS CHAPITAS
2	MANDALA	COSME Y DAMIAN	FENIX	LA DESCOCADA	LOS DUENDES	JADE	DIABLITOS VERDES	OHANA	DON BARULLO	ZODIACO
3	YOGUIS	DESCARA2	LOS TOBY'S	S.C.	DEPORTADOS	COLIBRIQUIS	HEREDEROS	ZERO	CHERRY'S	FRENESI
4	QUIJOTES	DULCINEA	MANO A MANO	WONKA'S	ADRENALINA	LA ZAFADA	SAPHIRUS	LOS PEPINITOS	LOS BUSCADORES	BUBY'S BIS
Prom. por etapa	**5,44**	**5,43**	**5,42**	**5,36**	**5,44**	**5,38**	**5,43**	**5,40**	**5,42**	**5,42**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 12: Segundo fixture alternativo, (Modelo 202).

Analizando en conjunto con ADICAPRO se elabora un tercer fixture alternativo. Es aquí donde resulta pertinente una variación del modelo base. En este sentido se consideró que el cierre de la primera ronda se programara en función de los resultados de la edición del año anterior, imponiendo que el orden de actuación fuera de la siguiente forma: las agrupaciones que habían obtenido los terceros puestos actuarían antes que las que habían obtenido el segundo y primer puesto, y análogamente las agrupaciones que habían obtenido el segundo puesto actuarían antes de las que habían obtenido el primer puesto. Vale aclarar que, en el caso de la categoría Revistas, son cuatro las agrupaciones de cierre de etapas, por lo que también se contempló el cuarto puesto de la edición 2020.

Las agrupaciones habilitadas para actuar en el cuarto horario pertenecían a las siguientes categorías: Parodistas, Revistas y Murgas. Teniendo esto en cuenta es que ADICAPRO solicitó que en esta variante se alternaran las categorías en las etapas de modo que el fixture quedara equilibrado también en este sentido. Esto implicaba que, si en una etapa cerraba una agrupación de la categoría parodistas, por ejemplo, en las dos etapas siguientes debían cerrar agrupaciones de Revistas y Murgas. Lo mismo sucedería si en una etapa cerraba una agrupación de la categoría Murgas, no así en caso de ser la categoría Revistas, ya que las agrupaciones habilitadas para cerrar las etapas de esta categoría son cuatro. Esto se traduce en que iba a existir una sola etapa en la que le correspondería actuar a una agrupación de esta categoría, seguida en la etapa consecutiva de una agrupación de Murgas o Parodistas, para luego volver a actuar otra agrupación de Revistas. A esta variante del caso base se le denominó “Modelo 204”.

Se listan en la Tabla 3 las agrupaciones que actuarían en el horario 4 y los puestos que obtuvieron en la edición 2019 del Carnaval.

Tabla 3: Agrupaciones asignadas para actuar en el cuarto horario y los puestos obtenidos en la edición 2019 del Carnaval.

CATEGORÍA	AGRUPACIÓN	PUESTO EN LA EDICIÓN 2019
Murgas	La zafada	1
Murgas	Los pepinitos	2
Murgas	Mano a mano	4
Parodistas	Buby's bis	1
Parodistas	Quijotes	2
Parodistas	Wonka's	3
Revistas	Dulcinea	1
Revistas	Saphirus	2
Revistas	Adrenalina	3
Revistas	Los buscadores	4

Nótese que en el caso de la categoría Murgas no se menciona la agrupación que obtuvo el tercer puesto: esto se debe a que esta agrupación no participó del Concurso en la edición de 2020.

Se muestra en la Figura 13 el fixture resultante teniendo en cuenta las nuevas restricciones.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	COLIBRIQUIS	CHERRY'S	FRENESI	DESCARA2	ZERO	LOS TOBY'S	HEREDEROS	DEPORTADOS	S.C.	YOGUIS
2	JADE	LA DESCOCADA	LOS DUENDES	COSME Y DAMIAN	OHANA	FENIX	ZODIACO	DON BARULLO	MANDALA	DIABLITOS VERDES
3	TROYANOS	GNOMOS	BAM BAM	PRINCIPES	CACHIRULOS	LOS CHAPITAS	ZABRITOS	CELESTINOS	VALU'S	TOON'S
4	MANO A MANO	LOS BUSCADORES	WONKA'S	LOS PEPINITOS	ADRENALINA	QUIJOTES	LA ZAFADA	SAPHIRUS	BUBY'S BIS	DULCINEA
Prom. por etapa	**5,42**	**5,36**	**5,41**	**5,40**	**5,42**	**5,36**	**5,47**	**5,46**	**5,38**	**5,48**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	BAM BAM	TROYANOS	PRINCIPES	LOS CHAPITAS	CACHIRULOS	CELESTINOS	VALU'S	ZABRITOS	TOON'S
2	LA DESCOCADA	COSME Y DAMIAN	JADE	LOS DUENDES	FENIX	OHANA	DON BARULLO	MANDALA	ZODIACO	DIABLITOS VERDES
3	CHERRY'S	FRENESI	COLIBRIQUIS	DESCARA2	LOS TOBY'S	ZERO	DEPORTADOS	S.C.	HEREDEROS	YOGUIS
4	LOS BUSCADORES	WONKA'S	MANO A MANO	ADRENALINA	QUIJOTES	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LA ZAFADA	DULCINEA
Prom. por etapa	**5,36**	**5,46**	**5,42**	**5,37**	**5,36**	**5,40**	**5,46**	**5,38**	**5,47**	**5,48**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 13: Tercer fixture alternativo, (Modelo 204). Restricciones en base a los resultados del concurso en la edición 2019.

Finalmente, para poder proporcionarle dos calendarios más a ADICAPRO y que de esta forma se pudiera contar con variedad de opciones al momento de elegir el fixture definitivo, se crearon dos variantes del Modelo R2, (caso base), imponiendo aleatoriamente que dos de las agrupaciones no actuaran en los horarios y etapas asignados en el fixture resultante del modelo R2. A estos modelos se les denominaron “Modelo 205” y “Modelo 206”, respectivamente. De esta forma se obtuvieron los calendarios que se muestran en las Figuras 14 y 15.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	YOGUIS	DESCARA2	FRENESI	HEREDEROS	LOS TOBY'S	DEPORTADOS	COLIBRIQUIS	CHERRY'S	S.C.	ZERO
2	MANDALA	COSME Y DAMIAN	DIABLITOS VERDES	LA DESCOCADA	FENIX	LOS DUENDES	OHANA	ZODIACO	JADE	DON BARULLO
3	TROYANOS	PRINCIPES	CACHIRULOS	CELESTINOS	ZABRITOS	TOON'S	GNOMOS	LOS CHAPITAS	VALU'S	BAM BAM
4	MANO A MANO	SAPHIRUS	BUBY'S BIS	ADRENALINA	LOS PEPINITOS	DULCINEA	WONKA'S	LA ZAFADA	QUIJOTES	LOS BUSCADORES
Prom. por etapa	**5,21**	**5,45**	**5,43**	**5,46**	**5,23**	**5,46**	**5,44**	**5,49**	**5,47**	**5,51**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	TROYANOS	CACHIRULOS	ZABRITOS	PRINCIPES	CELESTINOS	LOS CHAPITAS	VALU'S	TOON'S	GNOMOS	BAM BAM
2	MANDALA	COSME Y DAMIAN	LA DESCOCADA	FENIX	DIABLITOS VERDES	ZODIACO	JADE	LOS DUENDES	OHANA	DON BARULLO
3	YOGUIS	DESCARA2	FRENESI	LOS TOBY'S	HEREDEROS	CHERRY'S	S.C.	DEPORTADOS	COLIBRIQUIS	ZERO
4	MANO A MANO	BUBY'S BIS	SAPHIRUS	LOS PEPINITOS	ADRENALINA	LA ZAFADA	QUIJOTES	DULCINEA	WONKA'S	LOS BUSCADORES
Prom. por etapa	**5,21**	**5,41**	**5,34**	**5,43**	**5,40**	**5,49**	**5,47**	**5,46**	**5,44**	**5,51**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 14: Cuarto fixture alternativo, (modelo 205).

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	CHERRY'S	S.C.	ZERO	COLIBRIQUIS	DESCARA2	DEPORTADOS	LOS TOBY'S	HEREDEROS	YOGUIS	FRENESI
2	DULCINEA	MANDALA	LA DESCOCADA	JADE	LOS DUENDES	COSME Y DAMIAN	OHANA	DON BARULLO	FENIX	DIABLITOS VERDES
3	LOS CHAPITAS	VALU'S	BAM BAM	TROYANOS	CELESTINOS	TOON'S	GNOMOS	ZABRITOS	CACHIRULOS	PRINCIPES
4	LA ZAFADA	BUBY'S BIS	LOS BUSCADORES	MANO A MANO	ADRENALINA	LOS PEPINITOS	QUIJOTES	DULCINEA	WONKA'S	SAPHIRUS
Prom. por etapa	**6,15**	**5,38**	**5,26**	**5,42**	**5,48**	**5,47**	**5,44**	**5,42**	**5,31**	**5,47**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	BAM BAM	LOS CHAPITAS	VALU'S	TROYANOS	CELESTINOS	TOON'S	GNOMOS	ZABRITOS	CACHIRULOS	PRINCIPES
2	LA DESCOCADA	ZODIACO	MANDALA	JADE	LOS DUENDES	COSME Y DAMIAN	OHANA	DON BARULLO	FENIX	BAM BAM
3	ZERO	CHERRY'S	S.C.	COLIBRIQUIS	DESCARA2	DEPORTADOS	LOS TOBY'S	HEREDEROS	YOGUIS	VALU'S
4	LOS BUSCADORES	LA ZAFADA	BUBY'S BIS	MANO A MANO	ADRENALINA	LOS PEPINITOS	QUIJOTES	DULCINEA	WONKA'S	SAPHIRUS
Prom. por etapa	**5,26**	**5,49**	**5,38**	**5,42**	**5,48**	**5,47**	**5,44**	**5,42**	**5,31**	**6,10**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 15: Quinto fixture alternativo, (Modelo 206).

Los fixtures expuestos anteriormente se presentaron a ADICAPRO en noviembre y diciembre del año 2020 siendo el tercer calendario presentado, (modelo 204), el que la Asociación entendió que era el adecuado para desarrollar el concurso.

Luego de varias situaciones de incertidumbre, el 8 de diciembre se define que, debido a la situación sanitaria, el Carnaval finalmente no se realizaría. A pesar de esto, se trabajó en conjunto con ADICAPRO según lo previsto, tal como si se fuera a llevar a cabo el Carnaval.

Todos los fixtures anteriores se hicieron en base a un promedio erróneo, (4,75), en vez de 5,41 que es el promedio real. Se muestran estos calendarios porque fueron los que se le propusieron a ADICAPRO. La experimentación que se presentará en la Sección 4 sí se realizó en base a los promedios reales.

3.4. Modelado con Programación con Restricciones

El modelado mediante esta técnica consiste en representar el problema mediante la declaración de restricciones y encontrar soluciones que las satisfagan.

La selección de herramientas informáticas que permitan modelar y resolver este tipo de problemas es de gran importancia. Luego del relevamiento y estudio realizado para el Estado del Arte, (Anexo 1), se optó por Choco (4.10.6). Este fue desarrollado en 1999, es una iniciativa francesa, de código abierto diseñada especialmente para la investigación y el aprendizaje académico. Es una librería de Java dedicada a la Programación con Restricciones lo cual también se tuvo en cuenta para la selección.

La representación de los modelos descritos en la Sección 3.3 mediante Programación con Restricciones requirió introducirse en el lenguaje Java y la sintaxis utilizada en Choco para la representación de las restricciones.

Se utilizó una PC con las siguientes características: procesador Intel Core i7 - 5500U CPU, 2.40GHz y 8GB de memoria RAM, para el modelado se utilizó NetBeans 12.3, este es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java se importaron las librerías utilizadas por Choco. En la Figura 16 se pueden observar algunas de ellas.

```
package choco;

import org.chocosolver.solver.Model;
import org.chocosolver.solver.*;
import org.chocosolver.sat.*;
import org.chocosolver.util.*;
import org.chocosolver.memory.*;
import static org.chocosolver.solver.search.strategy.Search.*;
import org.chocosolver.solver.search.strategy.assignments.DecisionOperator;
import org.chocosolver.solver.search.strategy.selectors.values.*;
import org.chocosolver.solver.search.strategy.selectors.variables.*;
import org.chocosolver.cutoffseq.AbstractCutoffStrategy.*;
import org.chocosolver.solver.constraints.*;
import org.chocosolver.solver.variables.*;
import org.chocosolver.util.tools.ArrayUtils;
import org.chocosolver.util.objects.setDataStructures.iterable.IntIterableRangeSet ;
import static org.chocosolver.solver.Model.MINIMIZE;
import static org.chocosolver.util.tools.StatisticUtils.sum;
import org.chocosolver.solver.Solver ;

import java.util.Arrays;
import java.util.AbstractSet;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.Collections;
import java.io.*;
import javax.swing.*;
import java.math.*;
import java.io.FileWriter;
import java.io.File;
import java.io.IOException;
```

Figura 16: Librerías Choco.

Se entendió necesario comenzar la experimentación en Choco con el modelo de una ronda, (Modelo R1), el cual permitió obtener un primer modelo simplificado e ir validando las restricciones y los resultados obtenidos.

Luego el Modelo R2, Modelo 204 y 201 mencionados en la Sección 2.

3.4.1. Implementación en Choco

El problema de la confección del fixture tiene como objetivo asignar las distintas agrupaciones en cada uno de los horarios correspondientes a cada etapa de forma que la desviación entre el promedio de las ponderaciones por etapa y la ponderación promedio de las agrupaciones sea mínima, con el objetivo de balancear la concurrencia del público en cada una de las etapas.

A continuación, se muestran las características más relevantes del modelado en Choco.

3.4.1.1. Modelo

El objeto *model* es el componente clave, esta debe ser la primera instrucción, ya que es necesaria para declarar variables, restricciones, almacena las variables declaradas, las restricciones publicadas y da acceso a Solver.

Una vez creado el *modelo* como se puede observar en la Figura 17, se pueden definir las variables y restricciones.

```
// 2.Crear Modelo
    Model model = new Model("Carnaval de las promesas");
```

Figura 17: Crear Modelo.

3.4.1.2. Clase

La clase *Agrupacion* es una plantilla que define la forma del objeto agrupación; en ella se crean 4 atributos como se puede observar en la Figura 18, estos tienen que ver con el nombre de la agrupación, su ponderación, el horario en el cual pueden actuar en la primera ronda y la categoría a la que corresponde.

```
public class Agrupacion {

public String nombre;    // nombre agrupación
    public int pondera;      // ponderación de agrupación
    public int horariopos;  //  horario que puede actuar
    public String categagrup;  //  categoría de la agrupación
```

Figura 18: Clase Agrupacion.

3.4.1.3. Constantes

La cantidad de agrupaciones, etapas, horarios, rondas, categorías son algunas de las constantes que se pueden ver en la Figura 19, necesarias para la implementación del modelo en donde cada una de ellas es definida especificando el tipo de dato, nombre de la constante y el valor que toma.

```
int A = 40;    //cantidad de agrupaciones
int D = 10;    //cantidad de etapas por ronda
int T = 4;     // cantidad de horarios
int C = 2;    .//cantidad de ruedas
int K = 5;     // cantidad de categorias
int R = 8;     // cantidad de revistas
```

Figura 19: Declaración de Constantes Choco.

3.4.1.4. Variables

A continuación, se detallan algunas de las variables más importantes:

- Las agrupaciones del tipo *Agrupacion* son definidas, como se puede observar en la Figura 20, mediante un ciclo iterativo.

```
// Definir agrupaciones, categorías, horarios posibles en los que pueden actuar en primera ronda.
    Agrupacion l[] = new Agrupacion[A];
        for (int i = 0; i < l.length; i++)  {
            l[i] = new Agrupacion();
                                            }
```

Figura 20: Variable del tipo *Agrupacion*.

- Para la representación de la variable de decisión *X[i][d][t][c]* se entendió necesario modelar como una matriz de 4 dimensiones. En la Figura 21 se puede observar la construcción de la misma, así como también la definición del dominio mediante un ciclo iterativo definiendo los valores que puede tomar la variable.

```
// Variable de decisión. vale 1 si la agrupación i actua en el horario T en la etapa D en la ronda C, 0 si no.

   IntVar[][][][] X = new IntVar[A][D][T][C];
      for (int i = 0; i < A; i++) {
          for (int t = 0; t < T; t++)     {
              for (int d = 0; d < D; d++) {
                     for (int c = 0; c < C; c++) {
                     X[i][d][t][c] = model.intVar("X" + (i + 1) + "," + (d + 1) + "," + (t + 1)+ "," + (c + 1), new int[]{0, 1});
                                                 }
                                        }
                                        }
                               }
```

Figura 21: Variable de decisión.

- Las ponderaciones promedio por etapa se definen mediante dos arreglos *y_pond1* e *y_pond2* correspondientes a cada ronda, como se puede observar en la Figura 22. Esta ponderación estará sujeta a la variable de decisión *X[i][d][t][c]*.

```
// varibable de decisión y que mide la ponderación promedio por etapa.

 IntVar[] y_pond1 = new IntVar[10];
        for (int i = 0; i < 10; i++) {
            y_pond1[i] = model.intVar("y_pond1_" + (i+1), 0, 99999, true);
                                     }
 IntVar[] y_pond2 = new IntVar[10];
        for (int i = 0; i < 10; i++) {
            y_pond2[i] = model.intVar("y_pond2_" + (i+1), 0, 99999, true);
                                     }
```

Figura 22: Ponderaciones por etapa.

3.4.1.5. Restricciones

El modelado de las restricciones requirió un estudio riguroso de la documentación proporcionada por Choco, [24].

- El método *post()* en la restricción se utiliza para que el solver tome la misma en el proceso de solución.
- *model.arithm* se utiliza para expresar relaciones aritméticas entre dos o tres variables, o entre una o dos variables y una constante entera. Los operadores matemáticos para las restricciones de este tipo son: ">=", ">", "<=", "<", "=" y "!=". Por ejemplo: $model.arithm(x, " >= ", 4).post();$.
- *model.sum* se utiliza para sumar variables. Por ejemplo: $model.sum(vars, op, x).post();$.
- *model.scalar* permite hacer una suma de productos de elementos de dos arreglos.

A continuación, se detallan algunas de las restricciones más importantes necesarias para la implementación de los modelos.

Se comenzó modelando el calendario con dos restricciones básicas: la primera impone que se asigne una única agrupación en cada ventana de tiempo, es decir, en cada uno de los horarios correspondientes a cada etapa/ronda. La segunda impone que en cada ronda la asignación de una agrupación sea única.

- Mediante un ciclo iterativo se construye el arreglo *resultado*, este almacena para cada agrupación y ronda la variable *X[i][d][t][c]* instanciada en cada etapa y horario. Utilizando $model.sum$ se restringe la suma de los elementos de este arreglo a 1 como se puede ver en la Figura 23, es decir, que para cada agrupación en cada ronda la asignación de la misma debe ser única.

```
IntVar[] resultado = new IntVar[A];

            for (int i = 0; i < A; i++)     {               // agrupaciones
                for (int c = 0; c < C; c++) {               //  rondas
                    for (int d = 0; d < D; d++) {           // etapas
                        for (int t = 0; t < T; t++) {       // horarios
                            resultado[d*4+t] = X[i][d][t][k];
                                                }
                                            }
                model.sum(resultado,"=",1).post();       // Restricción
                                        }
                                        }
```

Figura 23: Restricción de asignación única por ronda.

- En la Figura 24 se modela la restricción que impone que, dentro de cada ronda, para cada etapa y horario, la sumatoria de los elementos del arreglo *p1* que corresponden a los elementos $X[i][d][t][c]$ valga 1. De esta forma, se garantiza que la asignación en cada horario sea única.

```
//1. Para cada etapa/horario hay una unica asignación

    for (int c = 0; c < C; c++) {                          // rondas
        for (int d = 0; d < D; d++) {                      // etapas
            IntVar[] pl = new IntVar[A];
                for (int t = 0; t < T; t++)      {         // horario
                    for (int i = 0; i < A; i++) {          //agrupaciones
                        pl[i] = X[i][d][t][c];
                                                 }
                model.sum(pl, "=", 1).post();              // Restricción
                                                 }
                                    }
                                }
```

Figura 24: Restricción de asignación única por ronda/etapa/ horario.

En la formulación matemática se definen los conjuntos A_1, A_2, A_3 y A_4 que representan las agrupaciones que pueden participar en cada horario de la primera ronda. En Choco se definió además un atributo horario a nivel de la agrupación. Debido a esto no fue necesario trabajar con conjuntos.

- En el caso de las agrupaciones que podían participar en el primer horario de la ronda 1, se construye, como se puede observar en la Figura 25, la variable $h1$. Esta corresponde a un arreglo de 10 elementos que mediante un ciclo iterativo almacena para cada agrupación la variable de decisión *X[i][d][0][0]* y luego se impone para cada agrupación que la sumatoria de los elementos de esta variable sea igual a 1. Así es que se garantiza que cada una de estas agrupaciones participen en alguna etapa del primer horario de la ronda 1.

```
// 3.  Solamente actuan en primer horario de la primera ronda las agrupaciones donde l[i].horariopos==1
    IntVar[] hl = new IntVar[10];
        for (int i = 0; i < A; i++)     {               // agrupaciones
            if (l[i].horariopos==1)     {
                for (int d = 0; d < D; d++) {           // etapas
                    hl[d]= X[i][d][0][0];
                                        }
                    model.sum(hl,"=", 1).post();
                                        }
                                        }
```

Figura 25: Restricción que impone el horario en que debe actuar una agrupación.

- Una restricción que hace a la calidad del espectáculo es la alternancia de categorías dentro de cada etapa. En este sentido, como se observa en la Figura 26 para Murgas, se impone que no haya dos agrupaciones de esta categoría en horarios consecutivos. Esta consideración se realiza para todas las categorías.

```
for (int j = 0; j < D-1; j++)   {
    for (int u = 0; u < C; u++) {
        for (int i = 0; i < A; i++) {
            if (l[i].categagrup=="murgas")  {
                for (int k = 0; k < A; k++) {
                    if (l[k].categagrup=="murgas")  {
                        model.arithm(X[i][j][3][u], "+", X[k][j+1][3][u], "<=", 1).post();
                                                    }
                                            }
                                            }
                                    }
                            }
                            }
```

Figura 26: Restricción para alternar categorías.

- La familia de restricciones que se observan en la Figura 27 acotan la diferencia entre la ponderación promedio del total de agrupaciones *pond_prom*, *(w)*, y la ponderación en cada una de las rondas para cada una de las etapas, de esta forma *z_pond1* y *z_pond2*, correspondientes a cada ronda, quedan acotadas por el mayor desvío de esas ponderaciones con respecto a la ponderación promedio.

```
for (int i = 0; i < 10; i++)     {
    model.arithm(y_pond1[i], "-", pond_prom , "<=", z_pond1[0]).post();
    model.arithm( pond_prom , "-",y_pond1[i], "<=", z_pond1[0]).post();
    model.arithm(y_pond2[i], "-", pond_prom , "<=", z_pond2[0]).post();
    model.arithm( pond_prom , "-",y_pond2[i], "<=", z_pond2[0]).post();
                                }
```

Figura 27: Familia de restricciones que acotan la ponderación promedio por etapa.

Luego se define la variable *tot_y_pond* que corresponde a la suma de *z_pond1* y *z_pond2*, de esta forma *tot_y_pond* queda definida como la suma de las desviaciones más grandes correspondientes a cada ronda. Ver a continuación la Figura 28.

```
IntVar tot_y_pond = model.intVar("tot_y_pond", 0, 100000);
model.sum(ArrayUtils.append(z_pond1, z_pond2), "=", tot_y_pond).post();
```

Figura 28: Variable definida como la suma de los desvíos más grandes de cada ronda.

3.4.1.6. Función objetivo

La función objetivo busca minimizar *tot_y_pond*, la suma de *z_pond1* y *z_pond2* de forma tal que la diferencia más grande en cada una de las rondas entre la ponderación promedio y la ponderación por etapa sea mínima, así también acotando el resto de las desviaciones de cada etapa. Esto se muestra en la Figura 29.

```
model.setObjective(Model.MINIMIZE, tot_y_pond);
```

Figura 29: Función objetivo.

3.4.1.7. Solución

Para enumerar las soluciones del problema se utilizó el ciclo $while(solver.solve())$. Éste permitió tener visibilidad de las soluciones ya encontradas a medida que el solucionador entraba en el árbol de búsqueda e intentaba encontrar nuevas soluciones.

El proceso de optimización es el siguiente: cada vez que se encuentra una solución, el valor de la variable objetivo se almacena y se publica un “corte”. El corte es una restricción adicional que establece que la siguiente solución debe ser, (estrictamente), mejor que la actual. Para resolver un problema de optimización, debe especificar qué variable optimizar y en qué dirección.

Mediante el ciclo $while$ se impone que mientras el árbol de búsqueda no se haya completado siga explorando; en caso de que exista una mejor solución, la devuelve. La instrucción $solver.limitTime$ permite establecer acotar la búsqueda a un determinado límite de tiempo, lo cual permitió poder trabajar con 3.600 segundos y 15.000 segundos. Esto se muestra en la Figura 30.

```
model.setObjective(Model.MINIMIZE, tot_y_pond);
Solver solver = model.getSolver();

solver.showShortStatistics();
solver.limitTime("20000s");
while(solver.solve()){
```

Figura 30: Solución del problema.

El fixture resultante para el Modelo R2 obtenido con Choco se puede observar en la Figura 31.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	HEREDEROS	ZERO	DESCARA2	YOGUIS	S.C.	LOS TOBY'S	COLIBRIQUIS	DEPORTADOS	CHERRY'S	FRENESI
2	DON BARULLO	MANDALA	LA DESCOCADA	LOS DUENDES	ZODIACO	FENIX	DIABLITOS VERDES	OHANA	COSME Y DAMIAN	JADE
3	TOON'S	LOS CHAPITAS	CACHIRULOS	ZABRITOS	BAM BAM	CELESTINOS	PRINCIPES	GNOMOS	VALU'S	TROYANOS
4	SAPHIRUS	LOS PEPINITOS	BUBY'S BIS	ADRENALINA	WONKA'S	LA ZAFADA	DULCINEA	QUIJOTES	LOS BUSCADORES	LOS BUSCADORES
Prom. por etapa	**5,57**	**4,96**	**4,94**	**5,80**	**4,90**	**5,68**	**5,83**	**5,77**	**5,64**	**3,71**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	CACHIRULOS	LOS CHAPITAS	BAM BAM	TOON'S	ZABRITOS	CELESTINOS	VALU'S	TROYANOS	PRINCIPES	GNOMOS
2	DON BARULLO	LA DESCOCADA	MANDALA	ZODIACO	LOS DUENDES	FENIX	JADE	OHANA	COSME Y DAMIAN	DIABLITOS VERDES
3	YOGUIS	ZERO	S.C.	HEREDEROS	DESCARA2	LOS TOBY'S	FRENESI	COLIBRIQUIS	DEPORTADOS	CHERRY'S
4	BUBY'S BIS	SAPHIRUS	WONKA'S	LOS PEPINITOS	ADRENALINA	LA ZAFADA	QUIJOTES	DULCINEA	MANO A MANO	LOS BUSCADORES
Prom. por etapa	**5,81**	**5,76**	**3,97**	**5,47**	**5,18**	**5,68**	**5,74**	**5,53**	**5,71**	**5,30**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 31: Fixture resultante del modelo base de 2 rondas, (Modelo R2).

Luego, como fue explicado en la Sección 3.3, se desarrollan varias alternativas. Para el modelado con Programación con Restricciones se consideraron los Modelos 204 y 201.

Para el Modelo 201, tal como se explicó para el caso de Programación Lineal Entera en la Sección 3.3, se basó en los resultados de la edición 2020 del concurso. En la Figura 32 se puede observar el calendario obtenido.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	LOS TOBY'S	HEREDEROS	COLIBRIQUIS	FRENESI	DEPORTADOS	S.C.	CHERRY'S	YOGUIS	DESCARA2	ZERO
2	ZODIACO	COSME Y DAMIAN	DON BARULLO	MANDALA	LOS DUENDES	LA DESCOCADA	JADE	DIABLITOS VERDES	FENIX	OHANA
3	GNOMOS	TOON'S	TROYANOS	BAM BAM	ZABRITOS	CACHIRULOS	VALU'S	CELESTINOS	PRINCIPES	LOS CHAPITAS
4	WONKA'S	LOS PEPINITOS	LOS BUSCADORES	BUBY'S BIS	SAPHIRUS	QUIJOTES	MANO A MANO	ADRENALINA	LA ZAFADA	DULCINEA
Prom. por etapa	5,08	5,75	5,16	5,19	5,33	5,39	5,67	5,63	5,76	5,17

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	TOON'S	ZABRITOS	CACHIRULOS	TROYANOS	BAM BAM	CELESTINOS	PRINCIPES	LOS CHAPITAS	VALU'S
2	DON BARULLO	ZODIACO	LOS DUENDES	COSME Y DAMIAN	LA DESCOCADA	MANDALA	DIABLITOS VERDES	FENIX	OHANA	JADE
3	HEREDEROS	FRENESI	LOS TOBY'S	DEPORTADOS	COLIBRIQUIS	S.C.	YOGUIS	DESCARA2	ZERO	CHERRY'S
4	WONKA'S	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LOS BUSCADORES	QUIJOTES	DULCINEA	MANO A MANO	ADRENALINA	LA ZAFADA
Prom. por etapa	5,73	5,62	5,01	5,54	4,92	5,09	5,64	5,81	5,16	5,63

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 32: Fixture resultante del modelo base 201, (Modelo 201).

Para el Modelo 204 se tomaron las condiciones descritas anteriormente en 3.3, como se puede observar en la Figura 33 el resultado de este modelo.

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	CHERRY'S	FRENESI	YOGUIS	ZERO	COLIBRIQUIS	S.C.	LOS TOBY'S	DESCARA2	HEREDEROS	DEPORTADOS
2	COSME Y DAMIAN	JADE	MANDALA	DON BARULLO	OHANA	LA DESCOCADA	LOS DUENDES	FENIX	ZODIACO	DIABLITOS VERDES
3	GNOMOS	PRINCIPES	VALU'S	LOS CHAPITAS	ZABRITOS	CACHIRULOS	TROYANOS	CELESTINOS	BAM BAM	TOON'S
4	LOS BUSCADORES	MANO A MANO	WONKA'S	ADRENALINA	LOS PEPINITOS	QUIJOTES	SAPHIRUS	LA ZAFADA	BUBY'S BIS	DULCINEA
Prom. por etapa	5,83	4,83	4,78	5,97	5,09	5,39	5,42	5,88	5,97	4,98

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	VALU'S	ZABRITOS	PRINCIPES	LOS CHAPITAS	CACHIRULOS	TOON'S	BAM BAM	CELESTINOS	TROYANOS
2	COSME Y DAMIAN	MANDALA	JADE	LA DESCOCADA	DON BARULLO	OHANA	LOS DUENDES	ZODIACO	FENIX	DIABLITOS VERDES
3	CHERRY'S	YOGUIS	COLIBRIQUIS	FRENESI	LOS TOBY'S	ZERO	HEREDEROS	S.C.	DEPORTADOS	DESCARA2
4	LOS BUSCADORES	WONKA'S	MANO A MANO	ADRENALINA	QUIJOTES	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LA ZAFADA	DULCINEA
Prom. por etapa	5,83	4,78	5,01	5,50	4,91	5,40	5,75	5,85	6,00	5,12

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Figura 33: Fixture resultante del modelo base 204, (Modelo 204).

4. Experimentación numérica y análisis comparativo

En esta sección se presentan los escenarios e instancias de prueba, el proceso de validación de los distintos modelos y los resultados de la experimentación numérica tanto en Choco como en AMPL/CPLEX.

4.1. Escenarios e Instancias

Para esta etapa del proyecto, se eligieron 4 escenarios para realizar la experimentación numérica. El escenario 1 cuenta con una instancia, el escenario 2 cuenta con dos instancias al igual que el escenario 3, y finalmente, el escenario 4 cuenta con cuatro instancias. En todos los casos el tiempo límite de ejecución elegido fue de 3.600 segundos. Se muestra en lo que sigue la Tabla 4, con los diferentes escenarios e instancias que se utilizaron, con los datos y el objetivo de cada caso. Vale recordar que los conjuntos $A1$, $A2$, $A3$ y $A4$ corresponden a los conjuntos de agrupaciones habilitadas a actuar en los horarios 1, 2, 3 y 4 respectivamente.

Tabla 4: Escenarios e instancias elegidos para la experimentación numérica.

ESCENARIO	INSTANCIA	TIEMPO LÍMITE (seg)	DATOS - Ponderaciones	OBJETIVO
1	**1**	3600	Ponderaciones reales para todas las agrupaciones	Comparar evolución de valores objetivo del caso base en Choco y AMPL/CPLEX
2	**1**	3600	Ponderaciones reales para los conjuntos A2, A3 y A4; ponderación 10 para el conjunto A1	Estudiar como varían el valor objetivo, los tiempos de ejecución y los calendarios si el conjunto A1 tiene la misma ponderación
2	**2**	3600	Ponderaciones reales para los conjuntos A1, A2 y A3; ponderación 10 para el conjunto A4	Estudiar como varían el valor objetivo, los tiempos de ejecución y los calendarios si el conjunto A4 tiene la misma ponderación
3	**1**	3600	Ponderación 10 para A1, A2, A3 y A4	Estudiar como varían los tiempos de ejecución si las ponderaciones de todas las agrupaciones son altas e iguales
3	**2**	3600	Ponderación 1 para A1, A2, A3 y A4	Estudiar como varían los tiempos de ejecución si las ponderaciones de todas las agrupaciones son bajas e iguales
4	**1**	3600	Ponderaciones aleatorias para todas las agrupaciones	Estudiar como varían el valor objetivo, los tiempos de ejecución y los calendarios si todas las agrupaciones tienen la misma ponderación
4	**2**	3600	Ponderaciones aleatorias para todas las agrupaciones	Ídem anterior

4	**3**	3600	Ponderaciones aleatorias para todas las agrupaciones	Ídem anterior
4	**4**	3600	Ponderaciones aleatorias para todas las agrupaciones	Ídem anterior

En el Anexo 3 se presentan los datos de cada caso.

4.2. Proceso de validación

Luego de implementar los modelos en Choco, se realizó una instancia de validación entre ambas técnicas de optimización, Programación Lineal Entera y Programación con Restricciones.

Como primer paso se validó que cada fixture obtenido con Programación con Restricciones cumpliera con las restricciones de la formulación matemática desarrollada en la Sección 3.2, y en el caso de los fixtures alternativos, que cumplieran también con las restricciones adicionales. Esta primera instancia se realizó verificando manualmente que los fixtures cumplieran con las restricciones especificadas en los modelos implementados. La misma se realizó sin inconvenientes, es decir, no se detectaron inconsistencias sobre el cumplimiento de las restricciones.

El segundo paso que se realizó fue validar los calendarios obtenidos a partir de Programación con Restricciones en el modelo de Programación Lineal Entera implementado en AMPL/CPLEX: esto consistió en ingresar en los códigos escritos para AMPL/CPLEX nuevas restricciones que detallaban las agrupaciones que debían actuar en todas las etapas de ambas rondas, en cada horario. De esta forma se verificaron dos cosas: la primera, que los calendarios obtenidos a partir de Choco fueran una solución factible en AMPL/CPLEX, y la segunda, verificar rápidamente que el valor objetivo era igual en ambos solvers.

En este punto fue donde se encontraron diferencias entre ambos modelos. En primer lugar, se encontró que se había definido de forma errónea la familia de restricciones (18): ésta aseguraba que en todas las etapas actuara al menos una agrupación de humoristas. El incumplimiento de esta restricción había pasado desapercibido hasta este momento y en este punto fue donde se detectó. En segundo lugar, se observó que ingresando los valores resultantes del tercer fixture alternativo obtenido en Choco, AMPL/CPLEX devolvía que no era una solución factible. Así es que se encontró que en el código ingresado en Choco se había tenido en cuenta la alternancia de categorías en el cierre de etapas solamente teniendo en cuenta dos etapas consecutivas en vez de tres.

Luego de corregir estos errores, se realiza la etapa 3 de validación. En este punto se logró validar todos los modelos tanto de Programación Lineal Entera como de Programación con Restricciones de forma independiente, pudiendo avanzar así con la experimentación numérica.

Se entendió pertinente comenzar con el Modelo R1. Esto permitió realizar una primera validación del modelado en Choco mediante Programación con Restricciones; es importante destacar que esto suponía un nuevo desafío por desconocer la herramienta. Este modelo considera las restricciones descritas en la formulación matemática en la Sección 3.2, pero teniendo en cuenta una ronda: es decir que las restricciones que se aplican a las dos rondas no fueron consideradas.

Se seleccionaron 2 instancias, escenario 1 e instancia 1 y escenario 4 e instancia 1, ya que de esta forma se aseguraba validar el calendario correspondiente a los modelados de una ronda,

(ver Anexo 3), tanto con ponderaciones reales como con ponderaciones aleatorias. La validación independiente, (validación de los modelos obtenidos en cada solver), consistió en imprimir los calendarios de estas dos instancias obtenidas en ambos solvers verificando de forma manual que para las dos técnicas cumplieran con todas las restricciones modeladas.

Nuevamente para estas 2 instancias se imponen en AMPL/CPLEX los valores obtenidos para las variables $x_{i,d,t,c}$, con el objetivo de verificar que los calendarios obtenidos a partir de Choco fueran factibles en AMPL/CPLEX y a su vez que la función objetivo fuera la misma en ambos solvers.

En este sentido se logró validar las soluciones de estas dos instancias mediante las etapas antes descritas sin arrojar inconsistencias. Luego se verificó en forma manual por separado el resto de las instancias arrojadas por ambos solvers.

A continuación, en la Tabla 5 se puede observar el registro de esta validación.

Tabla 5: Validación Modelo R1.

		VALIDACIÓN				
MODELO RONDA 1		PRIMERA ETAPA		SEGUNDA ETAPA	TERCERA ETAPA	
ESC.	INST.	CHOCO	AMPL/CPLEX	CHOCO EN AMPL/CPLEX	VERIFICACIÓN EN CHOCO Y AMPL/CPLEX	OBSERVACIONES
1	1	✓	✓	✓	✓	Validado sin registrar inconsistencias
2	1	✓	✓	✓	✓	Validado sin registrar inconsistencias
2	2				✓	Validado sin registrar inconsistencias
3	1				✓	Validado sin registrar inconsistencias
3	2				✓	Validado sin registrar inconsistencias
4	1	✓	✓	✓	✓	Validado sin registrar inconsistencias
4	2				✓	Validado sin registrar inconsistencias
4	3				✓	Validado sin registrar inconsistencias
4	4	✓	✓	✓	✓	Validado sin registrar inconsistencias

Para el modelo R2, 201, 204 se procedió de la misma forma que con el Modelo R1 a diferencia de los Modelos 202,205,206 que no se modelaron con Programación con Restricciones y fueron validados de forma manual.

A continuación, se detalla este proceso.

Para el Modelo R2 nuevamente se seleccionaron 2 instancias, escenario 1 e instancia 1 y escenario 4 e instancia 1, (de aquí en adelante se denominarán con la nomenclatura 1.1 y 4.1), y se verificó que tanto en Choco como en AMPL/CPLEX cada calendario obtenido cumpliera con las restricciones modeladas. En esta primera etapa no surgieron inconvenientes.

Nuevamente para estas 2 instancias se imponen en AMPL/CPLEX los valores de las variables $x_{i,d,t,c}$ obtenidas en Choco, con el doble objetivo de verificar que los calendarios obtenidos a partir de Choco fueran una solución factible en AMPL/CPLEX y verificar rápidamente que la función objetivo fuera igual en ambos solvers.

Se comparan nuevamente y se verifican de manera independiente todas las instancias de forma exitosa. Se muestra en la Tabla 6 el proceso de validación de este modelo.

Tabla 6: Validación Modelo R2

		VALIDACIÓN				
MODELO RONDA 2		PRIMERA ETAPA		SEGUNDA ETAPA	TERCERA ETAPA	
ESC.	INST.	CHOCO	AMPL/CPLEX	CHOCO EN AMPL/CPLEX	VERIFICACIÓN EN CHOCO Y AMPL/CPLEX	OBSERVACIONES
1	1	✓	✓	✓	✓	Se detecta en la 2da. etapa que el modelo de Choco no cumplía con la familia de restricciones (18). Se corrige y se valida correctamente.
2	1	✓	✓	✓	✓	Validado sin registrar inconsistencias
2	2				✓	Validado sin registrar inconsistencias
3	1				✓	Validado sin registrar inconsistencias
3	2				✓	Validado sin registrar inconsistencias
4	1	✓	✓	✓	✓	Se detecta en la 2da. etapa que el modelo de Choco no cumplía con la familia de restricciones (18). Se corrige y se valida correctamente.
4	2				✓	Validado sin registrar inconsistencias
4	3				✓	Validado sin registrar inconsistencias
4	4	✓	✓	✓	✓	Se detecta en la 2da. etapa que el modelo de Choco no cumplía con la familia de restricciones (18). Se corrige y se valida correctamente.

Los Modelos 201 y 204 se validaron de igual forma que los descritos anteriormente. Para estos dos modelos se incorporó la variable $b[i,c]$ como fue explicado en la Sección 3.3: esto implicó que para el caso del Modelo 201 se incorporen las restricciones (35) y (36). En la primera etapa de validación el modelo mediante Programación Lineal Entera no arrojó diferencias cumpliendo con las restricciones de manera satisfactoria. Sin embargo, al realizar la verificación de los cronogramas obtenidos mediante Programación con Restricciones se detectó que la variable $b[i,c]$ estaba mal formulada, esto se detectó en el caso 1.1 debido a que no cumplía con la familia de restricciones (35).

Se formuló nuevamente esta variable y se corroboró que de manera independiente cada modelado satisfaciera las restricciones. Para estas dos instancias se imponen nuevamente en AMPL/CPLEX las valores solución de las variables $x_{i,d,t,c}$ obtenidas en Choco.

Luego de solucionar este inconveniente se compararon nuevamente y se verificaron de manera independiente el resto de las instancias de forma exitosa. Se muestra en la Tabla 7 el proceso de validación para este modelo.

Tabla 7: Validación Modelo 201.

		VALIDACIÓN				
MODELO 201		PRIMERA ETAPA		SEGUNDA ETAPA	TERCERA ETAPA	
ESC.	INST.	CHOCO	AMPL/CPLEX	CHOCO EN AMPL/CPLEX	VERIFICACIÓN EN CHOCO Y AMPL/CPLEX	OBSERVACIONES
1	1	✓	✓	✓	✓	Se detecta en la 2da. etapa que en el modelo de Choco la variable b[i,c] estaba definida incorrectamente. Se corrige y se valida correctamente.
2	1	✓	✓	✓	✓	Validado sin registrar inconsistencias
2	2				✓	Validado sin registrar inconsistencias
3	1				✓	Validado sin registrar inconsistencias
3	2				✓	Validado sin registrar inconsistencias
4	1	✓	✓	✓	✓	Se detecta en la 2da. etapa que en el modelo de Choco la variable b[i,c] estaba definida incorrectamente. Se corrige y se valida correctamente.
4	2				✓	Validado sin registrar inconsistencias
4	3				✓	Validado sin registrar inconsistencias
4	4	✓	✓	✓	✓	Se detecta en la 2da. etapa que en el modelo de Choco la variable b[i,c] estaba definida incorrectamente. Se corrige y se valida correctamente.

Es importante destacar que el error detectado en el Modelo 201 afectaba directamente sobre el Modelo 204 en Choco, este fue corregido antes de realizar la validación.

Para el Modelo 204 nuevamente se seleccionaron 2 instancias (1.1, 4.1), se verificó que tanto en Choco como en AMPL/CPLEX de manera independiente que cada calendario obtenido cumpliera con las restricciones definidas en el modelo algebraico. En esta primera etapa no surgieron inconvenientes.

Nuevamente, para estas dos instancias se imponen en AMPL/CPLEX las variables $x_{i,d,t,c}$ obtenidas como solución en Choco. En esta etapa se encontraron diferencias entre ambos modelos ya que la solución obtenida en Choco para ambas instancias no era factible en AMPL/CPLEX. A raíz de esto se detectó que en el modelado mediante Programación con Restricciones se había definido de forma errónea la familia de restricciones que diferencian este modelo del caso base: en el horario 4 se debían alternar las actuaciones de Parodistas, Revistas y Murgas. Luego de solucionar este inconveniente se verificaron de manera independiente el

resto de las instancias de forma exitosa y se comprobó la factibilidad de las soluciones de ambas instancias arrojadas por Choco en AMPL/CPLEX.

Se muestra a continuación en la Tabla 8 el registro de la validación del Modelo 204.

Tabla 8: Validación Modelo 204.

		VALIDACIÓN				
MODELO 204		PRIMERA ETAPA		SEGUNDA ETAPA	TERCERA ETAPA	
ESC.	INST.	CHOCO	AMPL/CPLEX	CHOCO EN AMPL/CPLEX	VERIFICACIÓN EN CHOCO Y AMPL/CPLEX	OBSERVACIONES
1	1	✓	✓	✓	✓	Se detecta en la 2da. etapa que el modelo de Choco no cumplía con las restricciones que aseguran la alternancia de Revistas, Parodistas y Murgas. Se corrige y se valida.
2	1	✓	✓	✓	✓	Validado sin registrar inconsistencias
2	2				✓	Validado sin registrar inconsistencias
3	1				✓	Validado sin registrar inconsistencias
3	2				✓	Validado sin registrar inconsistencias
4	1	✓	✓	✓	✓	Se detecta en la 2da. etapa que el modelo de Choco no cumplía con las restricciones que aseguran la alternancia de Revistas, Parodistas y Murgas. Se corrige y se valida.
4	2				✓	Validado sin registrar inconsistencias
4	3				✓	Validado sin registrar inconsistencias
4	4	✓	✓	✓	✓	Se detecta en la 2da. etapa que el modelo de Choco no cumplía con las restricciones que aseguran la alternancia de Revistas, Parodistas y Murgas. Se corrige y se valida.

Los Modelos 202, 205, 206 no arrojaron inconsistencias en el proceso de validación debido a que estos eran variantes a los antes descritos y ya habían sido validados previamente. De todas formas, se comprobó de forma manual que cada instancia de estos modelos cumpliera con el modelo algebraico.

Es importante destacar nuevamente que estos tres modelos solo se modelaron mediante Programación Lineal Entera.

4.3. Modelos

Los modelos seleccionados para la experimentación numérica fueron: Modelo R1, Modelo R2, Modelo 204 y Modelo 201.

En el caso de Modelo R1, se seleccionó debido a que fue un caso importante al momento de programar en Choco, con lo cual se entendió que era también importante la experimentación numérica en este caso. Además, es el único modelo de una ronda, por lo que resultaba interesante también estudiar cómo se comportaba en la experimentación numérica. Luego, el Modelo R2 se eligió entendiendo su importancia debido a que es el caso base de resolución. Los Modelos 201 y 204 fueron seleccionados dado que, en ambos casos fueron variantes propuestas por ADICAPRO. En particular, el Modelo 204 fue el que devolvió el fixture seleccionado finalmente para el concurso 2021 que finalmente fue suspendido.

A continuación, se analiza la experimentación de cada modelo en detalle.

4.3.1. Modelo R1 (Ronda 1)

Como se indicó anteriormente, este modelo es una versión simplificada del caso base, (Modelo Ronda 2), y es el único modelo que cuenta con una sola ronda.

Se muestra a continuación en la Tabla 9, los valores objetivo obtenidos en esta experimentación. En la misma se puede observar para los distintos casos, para AMPL/CPLEX y para Choco, el tiempo en encontrar la primera solución factible, el valor objetivo alcanzado en este tiempo, así como también el valor objetivo alcanzado en el tiempo límite, (3.600 segundos).

Tabla 9: Valores objetivo obtenidos para el Modelo Ronda 1.

		AMPL/CPLEX			CHOCO		
ESC.	**INST.**	**Tiempo en encontrar la 1era sol. Factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**	**Tiempo en encontrar la 1era sol. Factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**
1	**1**	0,31	6,37	0,12	6,57	7,65	0,69
2	**1**	0,44	6,29	1,23	6,62	5,10	1,70
2	**2**	0,3	5,82	0,25	7,14	6,22	0,81
3	**1**	0,14	0	-	8,12	0	-
3	**2**	0,06	0	-	8,52	0	-
4	**1**	0,55	10,62	0,13	6,46	12,65	0,47
4	**2**	0,30	9,27	0,11	6,75	14,34	0,46
4	**3**	0,31	8,31	0,53	7,04	7,31	0,89
4	**4**	0,11	9,26	0,16	6,65	6,99	0,57

Además, se muestra también en las Figuras 34 y 35 el comportamiento de los distintos solvers en un gráfico en función del tiempo teniendo en cuenta los escenarios e instancias 1.1 y 4.4.

Éstos fueron elegidos en el caso del 1.1, debido a que representa los datos de ponderación real, y el 4.4 representa datos aleatorios. Es interesante estudiar este último escenario siendo que representaría una situación genérica.

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	0,12	3,63
20	0,12	3,63
30	0,12	3,63
40	0,12	3,63
50	0,12	2,06
100	0,12	1,16
200	0,12	1,16
300	0,12	1,16
500	0,12	1,05
1000	0,12	0,69
1500	0,12	0,69
2000	0,12	0,69
2500	0,12	0,69
3000	0,12	0,69
3600	0,12	0,69

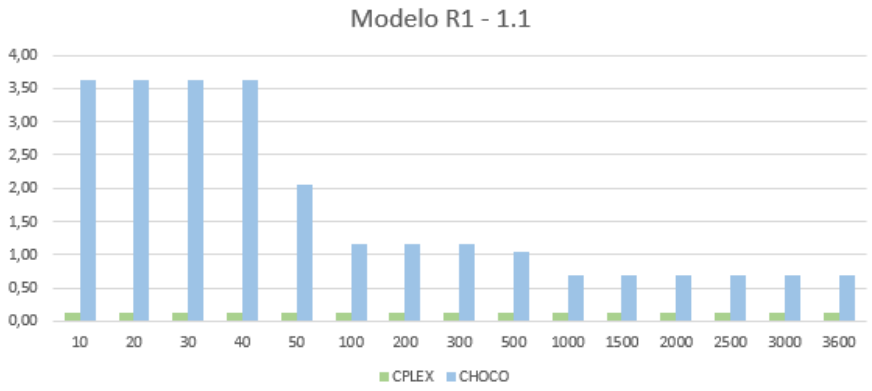

Figura 34: Gráfico de valor objetivo vs Tiempo del Modelo R1, caso 1.1.

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	0,29	2,19
20	0,16	1,11
30	0,16	1,11
40	0,16	1,11
50	0,16	1,11
100	0,16	1,11
200	0,16	1,11
300	0,16	1,11
500	0,16	0,98
1000	0,16	0,76
1500	0,16	0,57
2000	0,16	0,57
2500	0,16	0,57
3000	0,16	0,57
3600	0,16	0,57

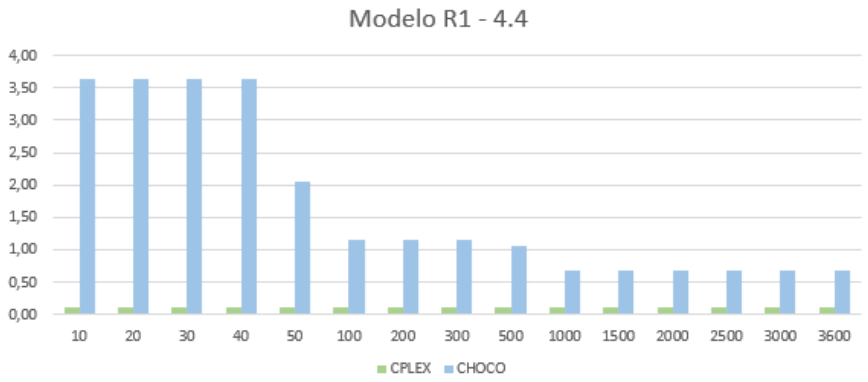

Figura 35: Gráfico de valor objetivo vs Tiempo del Modelo R1, caso 4.4.

Como puede observarse en la Tabla 9, solamente en 3 casos el valor objetivo de la primera solución factible es más bajo en el caso de Choco. En todos los casos el valor objetivo del tiempo límite es más bajo en AMPL/CPLEX, exceptuando el escenario 3 en la que ambos solvers encuentran la solución óptima como primera solución factible, casi de forma instantánea. Vale recordar que en el escenario 3 todas las ponderaciones son iguales: esto se traduce en que cualquier solución factible va a ser óptima, (el valor objetivo va a ser 0 en todos los casos).

Si se observa la Figura 34, en el caso de Choco se nota una mejora sustancial en el valor objetivo, pero aun así no alcanza el valor de AMPL/CPLEX. Notar que AMPL/CPLEX alcanza un valor mejor a los 10 segundos, (a pesar de no mejorarlo en ningún intervalo de tiempo), que Choco en todo el tiempo de ejecución. Análogamente, en el gráfico del caso 4.4, (Figura 35), se ve el mismo comportamiento de Choco: el valor objetivo disminuye sustancialmente, sin lograr alcanzar el

primer valor objetivo graficado de AMPL/CPLEX, aunque en este caso AMPL/CPLEX sí mejora el primer valor objetivo graficado.

4.3.2. Modelo R2 (Ronda 2)

Tal como se mencionó anteriormente, este modelo se considera el caso base con el cual se trabajó en una etapa inicial sin considerar variantes que tienen que ver con restricciones adicionales como se vio, por ejemplo, en los Modelos 204 y 201.

Se muestran en la Tabla 10 los resultados obtenidos en Choco y AMPL/CPLEX para los diferentes escenarios e instancias correspondientes al Modelo R2, y en las Figuras 36 y 37 la evolución de los valores objetivo de ambos solvers en el tiempo, nuevamente para los casos 1.1 y 4.4.

Tabla 10: Valores objetivo obtenidos para el Modelo Ronda 2.

		AMPL/CPLEX			CHOCO		
ESC.	**INST.**	**Tiempo en encontrar la 1era sol. factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**	**Tiempo en encontrar la 1era sol. factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**
1	**1**	1,61	13,26	0,24	2,00	8,79	7,81
2	**1**	6,47	9,78	2,46	1,37	8,84	7,51
2	**2**	1,50	15,26	0,50	1,84	10,38	2,25
3	**1**	0,05	0	-	1,35	0	-
3	**2**	0,06	0	-	1,33	0	-
4	**1**	2,00	15,28	0,26	*	*	*
4	**2**	1,64	13,50	0,22	15,21	11,93	4,59
4	**3**	3,06	204,06	1,16	1,39	17,09	3,55
4	**4**	1,66	12,03	0,32	2,53	17,92	9,49

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	0,60	7,96
20	0,60	7,96
30	0,60	7,96
40	0,60	7,96
50	0,60	7,96
100	0,58	7,96
200	0,37	7,81
300	0,28	7,81
500	0,28	7,81
1000	0,28	7,81
1500	0,24	7,81
2000	0,24	7,81
2500	0,24	7,81
3000	0,24	7,81
3600	0,24	7,81

Figura 36: Gráfico de valor objetivo vs Tiempo del Modelo R2, caso 1.1.

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	1,47	12,61
20	1,46	12,61
30	1,46	11,38
40	1,43	11,38
50	1,02	11,38
100	0,89	11,38
200	0,87	11,38
300	0,80	11,38
500	0,80	11,38
1000	0,70	11,38
1500	0,54	11,38
2000	0,54	9,49
2500	0,34	9,49
3000	0,34	9,49
3600	0,32	9,49

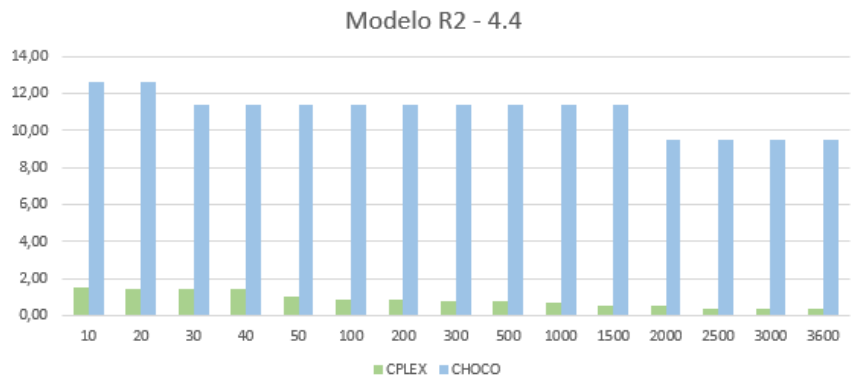

Figura 37: Gráfico de valor objetivo vs Tiempo del Modelo R2, caso 4.4.

Como se puede apreciar en la Tabla 10, los valores objetivo de las primeras soluciones factibles encontradas por AMPL/CPLEX son, en la mayoría de los casos, mayores que las encontradas por Choco, (contrariamente al caso anterior). Sin embargo, al mirar las Figuras 39 y 40, que representan la evolución del valor objetivo en función del tiempo, se concluye que en “el mediano/largo plazo” AMPL/CPLEX obtiene mejores resultados que Choco. También es importante destacar que en el escenario e instancia 4.1, se observa por primera vez que Choco no logra encontrar una solución factible en el tiempo límite, (3.600 segundos). Dada esta situación y para estudiar el comportamiento de esta y otras instancias, (1.1 y 4.4), en las cuales los valores objetivo eran mayores que los encontrados para este mismo modelo en las demás instancias, se ejecutaron estos escenarios/instancias durante 15.000 segundos. En primer lugar, en 4.1 se encontró que la primera solución factible se encuentra a los 13.204 segundos con un valor de 15,40, llegando al valor de 6,04 a los 13.235, sin variar luego de pasado este tiempo.

Por otro lado, se seleccionaron y se ejecutaron en Choco 15.000 segundos los casos 1.1 y 4.4. En el caso del 1.1 al ejecutar el modelo durante 15.000 segundos se logró encontrar una solución notoriamente mejor, correspondiente a un valor objetivo de 2,10 (7,81 a los 3600 segundos). Sin embargo, en el caso del 4.4 alcanzó la mejor solución a los 10.217 segundos con un valor de 8,27 (9,49 a los 3600 segundos), encontrando de esta forma que la mejora no es tan marcada.

Se puede observar que el modelo es sensible a los datos de entrada: en un caso la solución mejora sustancialmente a mayor tiempo de ejecución mientras que en el otro caso no varía demasiado.

Al observar las Figuras 36 y 37 se nota que las variaciones del valor objetivo tanto de Choco como de AMPL/CPLEX en 3.600 segundos no son tan sustanciales como las mencionadas anteriormente respecto al primer valor graficado, (valor alcanzado a los 10 segundos). De todas formas, sí es notoria la diferencia entre ambos valores objetivo, el de Choco y AMPL/CPLEX: los valores objetivo de AMPL/CPLEX representan tan sólo un 5%, en promedio, de los alcanzados por Choco.

4.3.3. Modelo 204

Se presentan para este modelo también los resultados de la experimentación en la Tabla 11, y los gráficos de evolución de valores objetivo en el tiempo, para los casos 1.1 y 4.4 en las Figuras 38 y 39, respectivamente.

Tabla 11: Valores objetivo obtenidos para el Modelo 204.

		AMPL/CPLEX			CHOCO		
ESC.	**INST.**	**Tiempo en encontrar la 1era sol. factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**	**Tiempo en encontrar la 1era sol. factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**
1	**1**	1,05	29,18	0,24	2,71	11,54	5,06
2	**1**	3,55	18,34	2,46	2,30	11,30	7,00
2	**2**	1,39	17,56	0,50	12,72	11,59	4,78
3	**1**	0,06	0	-	3,15	0	-
3	**2**	0,08	0	-	3,22	0	-
4	**1**	0,97	16,99	0,26	12,51	17,78	8,41
4	**2**	0,89	43,42	0,22	12,24	2,51	7,62
4	**3**	1,11	32,62	1,06	19,46	3,95	9,93
4	**4**	0,99	13,30	0,32	17,54	4,62	7,40

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	0,86	5,06
20	0,54	5,06
30	0,54	5,06
40	0,52	5,06
50	0,50	5,06
100	0,3	5,06
200	0,24	5,06
300	0,24	5,06
500	0,24	5,06
1000	0,24	5,06
1500	0,24	5,06
2000	0,24	5,06
2500	0,24	5,06
3000	0,24	5,06
3600	0,24	5,06

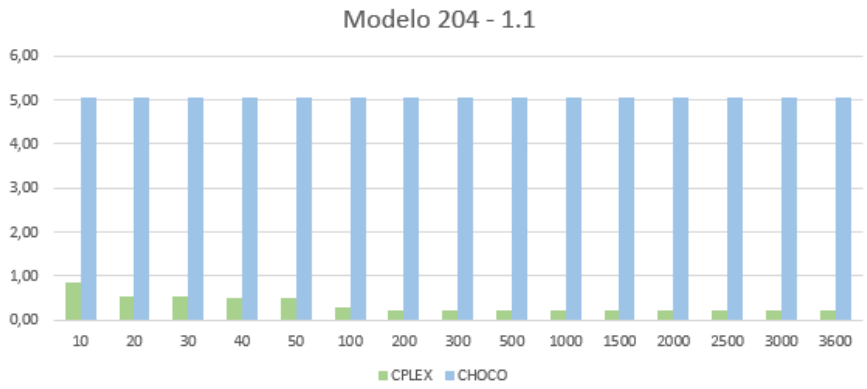

Figura 38: Gráfico de valor objetivo vs Tiempo del Modelo 204, caso 1.1.

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	1,57	9,17
20	1,57	9,17
30	1,42	8,35
40	1,40	8,35
50	1,40	8,35
100	0,77	7,40
200	0,64	7,40
300	0,60	7,40
500	0,56	7,40
1000	0,56	7,40
1500	0,34	7,40
2000	0,32	7,40
2500	0,32	7,40
3000	0,32	7,40
3600	0,32	7,40

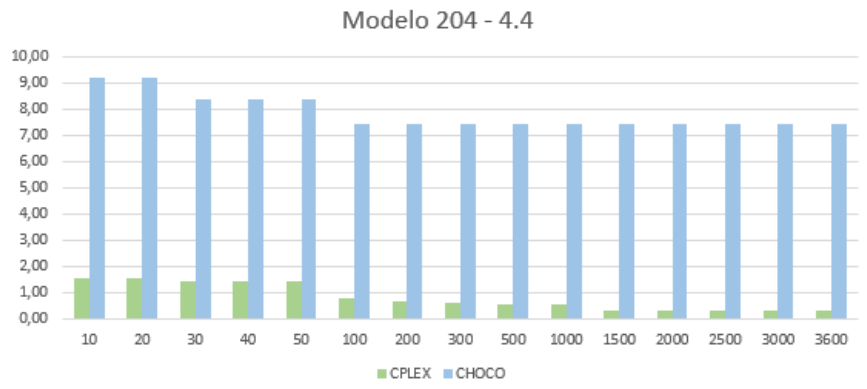

Figura 39: Gráfico de valor objetivo vs Tiempo del Modelo 204, caso 4.4.

En la experimentación con este modelo sucede algo muy particular: el valor objetivo más bajo alcanzado por Choco es de 5,06 mientras que los de AMPL/CPLEX al igual que en los otros modelos, están en el entorno de 0.20. De esta forma lo primero que se nota es que nuevamente AMPL/CPLEX logra mejores resultados que Choco en el tiempo límite. Es interesante notar, sin embargo, que en los valores objetivo hallados en la primera solución factible, pasa lo contrario. Si bien Choco encuentra los mismos, (o mejores), valores objetivo, en el mejor de los casos demora el doble de tiempo de ejecución. El valor objetivo hallado por Choco se mantiene constante a partir de los 10 segundos, durante todo el tiempo de ejecución siendo el mismo 6 veces mayor que el peor valor de AMPL/CPLEX: AMPL/CPLEX no sólo devuelve un valor objetivo menor a Choco en el primer tiempo graficado, si no que luego lo continúa mejorando mientras que Choco mantiene constante el primer valor hallado.

Al comparar el gráfico de la Figura 39 con el de la Figura 40, si bien los valores objetivo de Choco no se mantienen constantes como en el caso anterior, sucede algo similar al comparar los valores: en el mejor de los casos Choco devuelve un valor objetivo 5,8 veces mayor que la de AMPL/CPLEX. Si bien ambos solvers mejoran la primera solución graficada, (en el caso de Choco mejora 1,2 y AMPL/CPLEX casi 5 veces), la de AMPL/CPLEX una vez más es sustancialmente mejor.

Se seleccionaron los casos 1.1 y 4.1 de forma de establecer como tiempo límite de ejecución 15.000 segundos en Choco. Para el caso de 1.1 no se obtuvo una mejor solución en 15.000 segundos, manteniendo como mejor solución el valor objetivo correspondiente a 5,06 obtenido a los 11 segundos. El mismo comportamiento se da con el caso 4.1, el mejor valor objetivo lo logra a los 47,07 segundos y luego hasta los 15.000 segundos no logra encontrar uno mejor. Es importante destacar que en estos dos casos el solver finaliza la ejecución dado el valor del tiempo límite; es decir no termina de explorar el árbol de búsqueda.

4.3.4. Modelo 201

Finalmente, se muestran a continuación los resultados obtenidos para el Modelo 201 en la Tabla 12.

Tabla 12: Valores objetivo obtenidos para el Modelo 201.

		AMPL/CPLEX			CHOCO		
ESC.	**INST.**	**Tiempo en encontrar la 1era sol. factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**	**Tiempo en encontrar la 1era sol. Factible (seg)**	**Objetivo 1era sol. factible**	**Objetivo del tiempo límite**
1	**1**	1,63	23,24	0,24	*	*	*
2	**1**	4,42	18,34	2,46	2.367	8,66	6,14
2	**2**	1,47	17,56	0,50	2.464	11,06	6,03
3	**1**	0,02	0	-	**	**	**
3	**2**	0,03	0	-	***	***	***
4	**1**	2,25	2,22	0,26	2.786	12,93	7,03
4	**2**	1,59	28,68	0,30	1.186	15,29	4,07
4	**3**	1,59	32,62	1,06	2.826	20,72	10,77
4	**4**	1,73	12,58	0,56	3.573	15,30	13,44

Se presentan también los gráficos de evolución de valores objetivo en el tiempo en las Figuras 40 y 41.

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	1,21	0
20	1,21	0
30	1,21	0
40	1,21	0
50	1,12	0
100	0,92	0
200	0,48	0
300	0,44	0
500	0,28	0
1000	0,24	0
1500	0,24	0
2000	0,24	0
2500	0,24	0
3000	0,24	0
3600	0,24	0

Figura 40: Gráfico de valor objetivo vs Tiempo del Modelo 201, caso 1.1.

Tiempo (seg)	Valor objetivo	
	CPLEX	CHOCO
10	0,92	0
20	0,92	0
30	0,92	0
40	0,92	0
50	0,92	0
100	0,92	0
200	0,78	0
300	0,72	0
500	0,72	0
1000	0,72	0
1500	0,58	0
2000	0,58	0
2500	0,58	0
3000	0,56	0
3600	0,56	13,44

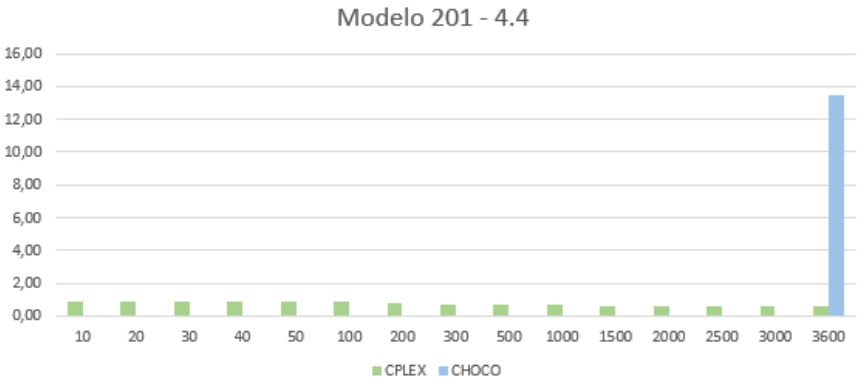

Figura 41: Gráfico de valor objetivo vs Tiempo del Modelo 201, caso 4.4.

Se encuentran en este modelo ciertos comportamientos de Choco diferentes a los vistos en la experimentación de los Modelos R1, R2 y 204. En primer lugar, observando la Tabla 12 se encuentra que los resultados de Choco para el caso 1.1 no se listan: esto es debido a que el solver no encuentra una solución factible en el tiempo límite de ejecución. Lo análogo sucede en los casos 3.1 y 3.2, en los cuales resulta muy peculiar dado que el escenario 3 corresponde a iguales ponderaciones para todas las agrupaciones, es decir, todas las agrupaciones ponderadas con los valores 1 o 10 para cada caso. Esto se traduce en que cualquier solución factible va a ser, a su vez, óptima. Estos casos se ejecutaron nuevamente con tiempos de ejecución mayores a los 3.600 anteriores, donde Choco encuentra soluciones a los 4.346 segundos en el caso 3.1, y 4.129 en el caso 3.2, (ambas óptimas, tal como se esperaba en ambos casos). En segundo lugar, en el caso 4.4 Choco encuentra la primera solución factible a los 3,573 segundos, comportamiento que hasta ahora no se había observado. Este comportamiento es el que determina que el gráfico de evolución del valor objetivo sea diferente a todos los demás, (Figura 44), al igual que sucede en el gráfico de la Figura 43, en donde también es diferente a la experimentación de los modelos R1, R2 y 204 por lo comentado anteriormente.

Debido a que los tiempos para encontrar la primera solución para este modelo son relativamente altos teniendo en cuenta los 3.600 segundos que se establecieron como tiempo límite de

ejecución, para poder determinar el comportamiento de Choco se seleccionaron nuevamente los casos 1.1 y 4.4.

Se ejecutó el caso 1.1 durante 15.000 segundos obteniendo como valor objetivo 3,38 a los 4,39 segundos, (10,17 a 4.137,98 segundos). Para el 4.4, (15,3 a 3.572,73 segundos), llega a un valor objetivo de 3,59 en 9.395,6 segundos. Como se puede observar en los dos casos, el tiempo de ejecución es de importancia logrando obtener valores objetivos sustancialmente menores.

Se destaca que no se encuentran similitudes entre los calendarios obtenidos por AMPL/CPLEX y Choco en los casos 1.1 y 4.1, en ninguno de los modelos, (ver Anexo 4).

4.4. Conclusiones de la experimentación numérica

En términos generales se puede concluir que AMPL/CPLEX logró encontrar una primera solución factible para todos los casos en un tiempo razonablemente bueno, en promedio de 1,25 segundos. Comparando este primer valor con la solución del tiempo límite podemos observar que distan mucho. El valor objetivo alcanzado en el tiempo límite, (3.600 segundos), mejora un 96,4% con respecto al primer valor objetivo hallado.

En el caso de Choco, si bien el tiempo promedio en encontrar la primera solución factible es de 480,60 segundos, esto se debe a que para el caso del Modelo 201 los valores son muy elevados, lo que incrementa notablemente el promedio. Sacando estos casos, el promedio de tiempo en hallar la primera solución factible en Choco disminuye a 6,80 segundos, lo cual es razonable considerando los tiempos de ejecución. En cuanto a la mejora del valor objetivo, el mismo solo mejora 27,3% en el tiempo límite con respecto al primer valor hallado.

Se concluye también que el tiempo límite de ejecución seleccionado fue adecuado ya que las soluciones encontradas por ambos solvers se estabilizan luego de los 1.500 segundos en la mayoría de los casos. Esto se puede observar en las Figuras 34 a 41. Sin embargo, en ambos solvers se podría obtener mejores soluciones con tiempos de ejecución más prolongados, ya que en ningún caso se alcanza la solución óptima. Es importante mencionar que para los casos en que se seleccionaron tiempos de ejecución de 15.000 segundos en Choco, tampoco se logró encontrar la solución óptima.

5. Conclusiones

En este informe se presenta el trabajo realizado sobre la planificación de espectáculos artísticos mediante Programación con Restricciones, aplicado a la configuración del calendario del Carnaval de las Promesas edición 2020/2021.

En primer lugar, con respecto al relevamiento y análisis de literatura se logró encontrar diversos artículos relativos a la técnica de Programación con Restricciones en diversas áreas por lo que se logra estudiar la aplicación de esta herramienta en distintos campos. Si bien no se encuentran trabajos de investigación en el ámbito artístico, sí se relevan trabajos de planificación de eventos en otras áreas como ser transporte, industria o académicos. Esta etapa fue parte del trabajo necesario para lograr realizar el modelado con Programación con Restricciones.

Con respecto al objetivo principal, en la primera etapa correspondiente al modelado mediante Programación Lineal Entera se logró proporcionar y validar con ADICAPRO distintos fixtures para la edición 2020/2021 del Carnaval. En una segunda etapa de implementación de los modelos con Programación con Restricciones, se logró obtener el conocimiento necesario para utilizar esta herramienta en el software seleccionado, (Choco). Si bien la implementación resultó dificultosa dado el conocimiento que la misma requería, se puede concluir que se lograron obtener los calendarios mediante esta herramienta también.

El desarrollo del modelado mediante Programación con Restricciones permitió también realizar una experimentación numérica que permitió comparar esta técnica con Programación Lineal Entera. Si bien ambos solvers utilizados, (AMPL/CPLEX y Choco), presentan comportamientos distintos, en términos generales se puede concluir que en el caso de estudio presentado en este trabajo, se encontró que se lograron mejores resultados con AMPL/CPLEX en términos de tiempos de ejecución y valores objetivo.

Como trabajos a futuro, se considera investigar la implementación de heurísticas que permitan obtener mejores resultados que los obtenidos con Choco, realizar una revisión del código implementado en Choco para así poder determinar si se pueden lograr mejores resultados. En este sentido resultaría interesante también investigar el comportamiento de otros solvers de Programación con Restricciones para comparar con los resultados obtenidos.

Para las dos técnicas de optimización, (Programación Lineal Entera y Programación con Restricciones), se implementaron modelos con características propias de la edición 2020/2021 del Carnaval de las Promesas, por lo que se considera que sería beneficioso implementar modelos independientes de la cantidad de agrupaciones, de las categorías y demás variables que hoy se encuentran dentro del código.

Finalmente, sería también de gran utilidad el desarrollo de una interfaz gráfica que permita mejorar aspectos de usabilidad, ingreso de datos de entrada y presentación de resultados, independizando a los usuarios de ADICAPRO de la Facultad de Ingeniería.

Bibliografía

[1] Definición de evento, Diccionario de la Real Academia Española. https://dle.rae.es/evento
Último acceso: 24/10/2021

[2] Definición de solver, Wikipedia. https://en.wikipedia.org/wiki/Solver
Último acceso: 24/10/2021

[3] Definición de planeamiento, Wikipedia. https://es.wikipedia.org/wiki/Planeamiento
Último acceso: 24/10/2021

[4] Networking.
https://networkingrd.net/2019/10/16/concepto-planificacion-que-es-y-para-que-sirve/
Último acceso: 24/10/2021

[5] G. Durán, S. Durán, J. Marenco, F. Mascialino, P. A. Rey. Programación matemática para los fixtures de los torneos profesionales del básquet de la Argentina en un formato NBA. *Revista Ingeniería de Sistemas Volumen XXX*, 35-61, 2016.

[6] A. Aggoun, A. Vazacopoulos. Solving Sports Scheduling and Timetabling Problems with Constraint Programming. *Economics, Management and Optimization in Sports*, 243-264, 2004.

[7] R. Russell, T. Urban. A constraint programming approach to the multiple-venue, sport-scheduling problem. *Computers & Operations Research 33*, 1895-1906, 2006.

[8] J. Bautista, M. Mateo, R. de la Torre. Programación Lineal Entera Mixta para asignar Voluntarios a la Organización de Eventos deportivos. Caso Aptitud no restringida, 2018.

[9] T. Benoist, E. Bourreau, B. Rottembourg. The TV-Break Packing Problem. *European Journal of Operational Research Volume 176*, 1371-1386, 2007.

[10] B. M. Smith, S. C. Brailsford, P. M. Hubbard, H. P. Williams. The progressive party problem: Integer linear programming and constraint programming compared. Principles and Practice of Constraint Programming. *Lecture Notes in Computer Science 976*, 36-52, 1995.

[11] A. Cardozo, J.M. Machin, C. Guido. Métodos cuantitativos para la programación de espectáculos artísticos: una aplicación en el carnaval uruguayo, 2020.

[12] R. Bai, J. Xie. Heuristic algorithms for simultaneously accepting and scheduling advertisements on broadcast television. *Journal of Information and Computing Science* 1, 245–251, 2006.

[13] A. Kimms, M. Muller-Bungart. Revenue management for broadcasting commercials: the channel's problem of selecting and scheduling the advertisements to be aired. *Journal of Revenue and Pricing Management 1*, 28–44, 2007.

[14] S. Bollapragada, H. Cheng, M. Phillips, M. Garbiras, M. Scholes, T. Gibbs, M. Humphreville. NBC's optimization systems increase revenues and productivity. *Interfaces 32*, 47–60, 2002.

[15] Programación con Restricciones, Wikipedia.
https://en.wikipedia.org/wiki/Constraint_programming
Último acceso: 20/09/2021

[16] F. Barber, M. Salido. Introducción a la Programación de Restricciones, *Revista Iberoamericana de Inteligencia Artificial 7,* 13-30, 2003.

[17] S. Lizama, A. Muñoz. Algoritmos de Consistencia para Programación con Restricciones, 2013.

[18] Sitio web de la Intendencia de Montevideo. https://montevideo.gub.uy/areas-tematicas/cultura-y-tiempo-libre/carnaval-y-llamadas/historia-del-carnaval
Último acceso: 3/10/2021.

[19] Universia, Portal de Universidades uruguayas. https://www.universia.net/uy/actualidad/vida-universitaria/breve-repaso-historia-carnaval-uruguayo-1148747.html
Último acceso: 29/09/2021.

[20] Artículo de TV Show sobre la murga Agarrate Catalina.
https://www.tvshow.com.uy/carnaval-agarrate-catalina-volvio-brillo-abrio-polemica-cuple.html
Último acceso: 29/09/2021.

[21] Sitio oficial de DAECPU, (Directores Asociados de Espectáculos Carnavalescos Populares del Uruguay). https://www.daecpu.org.uy/
Último acceso: 29/09/2021.

[22] Sitio oficial de ADICAPRO. www.adicapro.org
Último acceso: 1/10/2021.

[23] Reglamento del Carnaval de las Promesas, Sitio oficial de ADICAPRO.
http://adicapro.org/0documentos/Reglamento_Carnaval_Promesas_2016.pdf
Último acceso: 1/10/2021.

[24] Página oficial de Choco-Solver, tutoriales en línea. https://choco-solver.org/tutos/
Último acceso: 12/10/2021.

[25] J. A. Gutiérrez Quezada, A. E. López Estay. Tesis de grado, Escuela de Ingeniería Informática, Facultad de Ingeniería, Pontificia Universidad Católica de Valparaíso, 2011.

ANEXO 1 - Estado del arte

PROBLEMAS DE PLANIFICACIÓN MEDIANTE PROGRAMACIÓN CON RESTRICCIONES

ESTADO DEL ARTE

Autores:
Inés Baráibar
Esteban Machado

Tutores:
Pedro Piñeyro
Héctor Cancela

Resumen

Este documento tiene como objetivo central presentar el relevamiento y análisis de la literatura orientada a Problemas de Planificación aplicado en diversas actividades que tienen que ver con Asignación de Tareas, Asignación de Recursos, Planificación de calendarios entre otros. Se orientó la búsqueda en Programación con Restricciones para de esta forma poder determinar en qué etapa de desarrollo se encuentra esta área que resulta motivante investigar.
Comienza con una breve introducción a la Programación con Restricciones, destacando los principales conceptos, los algoritmos de búsqueda más conocidos como por ejemplo Generación y Prueba, Backtracking y Chequeo hacia Adelante. Se presentan Técnicas de consistencia como Consistencia de Nodo, Consistencia de Arco, finalizando con una breve introducción a las Estrategias de Enumeración.
Finaliza el documento con un análisis sobre las herramientas informáticas más utilizadas para resolver problemas mediante Programación con Restricciones, se presenta un relevamiento de las herramientas informáticas que se emplearon en los artículos mencionados en este informe, con la finalidad de que el lector pueda lograr un primer acercamiento a los software disponibles.

Palabras Clave: Programación con Restricciones, Problemas de Satisfacción de Restricciones, Problemas de Planificación, Optimización.

ÍNDICE

1. Introducción

La planificación es un proceso de toma de decisiones, el cual tiene como fin lograr determinados objetivos de forma ordenada. Para ello se debe tener en cuenta el contexto actual, los factores externos e internos que pueden imponernos condiciones influyendo en los resultados esperados. Planificar [1] supone analizar y estudiar los objetivos propuestos, así como la forma en la que vamos a conseguirlos. Es una herramienta de acción que nos permite decidir qué vamos a hacer y por qué. Tiene muchos beneficios, pero sobre todo clarifica muchas dudas, ayuda a determinar la necesidad de recursos para conseguir objetivos, clarifica las actividades y las dudas respecto a los objetivos buscados, cuantifica los niveles de desempeño para tener éxito, establece prioridades, evidencia debilidades y fortalezas para conseguir objetivos. Se puede entender como una actividad humana esencial para la supervivencia de una comunidad, la cual tenemos incorporada de una manera intrínseca en nuestra vida cotidiana para lograr nuestros objetivos.
En el ámbito laboral, la planificación de diversas actividades que se llevan a cabo tiene como objetivo alinear y coordinar de forma eficiente y eficaz los recursos materiales y el capital humano para obtener de forma ordenada los mayores beneficios económicos y a la vez poder ser competitivos. Dentro de esta área, los problemas de asignación más interesantes tienen que ver con la asignación de tareas, turnos de trabajo, maquinaria y recursos materiales entre otros. Si bien el fin último muchas veces tiene que ver con resultados económicos, estos problemas de asignación buscan directamente reducir tiempos muertos, tiempos de parada de maquinaria, organizar tareas y turnos de trabajo de forma adecuada y cumpliendo con las restricciones propias de cada problema.

Resulta interesante abordar la planificación de eventos. En este sentido, un evento se define como [2] “un suceso importante y programado, de índole social, académica, artística o deportiva”. La Programación de eventos refiere a la coordinación y planificación de estos eventos asignando el comienzo y finalización de cada uno de ellos considerando las restricciones propias de cada evento y otros factores que pueden tener que ver con la organización, reglamentaciones, términos económicos entre otros.

Dentro de la planificación de eventos deportivos, la planificación de las ligas deportivas es uno de los problemas más desafiantes; como por ejemplo, planificar el fixture de una liga asignando fechas, partidos, localías, giras entre otros factores.

En el ámbito académico la planificación de horarios de clase o exámenes, en el ámbito social y cultural la planificación de competencias, eventos o espectáculos. Cabe destacar que, en este rubro, el fin muchas veces no es solamente económico, siendo que incluso este factor directamente no es considerado cuando el mismo tiene como objetivo el bienestar social.
Dada la complejidad de la problemática a los que nos debemos enfrentar, al momento de planificar, en algunos casos llevaría mucho tiempo resolverlos de forma manual por lo tanto es necesario recurrir a técnicas matemáticas y solvers que permitan abordar estos problemas y obtener soluciones óptimas de forma rápida en términos de tiempos de ejecución. Dentro de las técnicas de optimización matemática se destacan la Programación Matemática, (Programación Lineal, Programación Entera, Programación Entera-Mixta entre otras), y la Programación con Restricciones.

A continuación, el documento se desarrolla de la siguiente forma: en la Sección 2 se presenta una breve introducción a la Programación con Restricciones; en la Sección 3 se desarrolla el análisis de la literatura relevada; en la Sección 4 se desarrolla una revisión de los solvers; en la Sección 5 se presentan las conclusiones.

2. Programación con Restricciones

Muchos de los problemas a los cuales nos enfrentemos en nuestra vida cotidiana son particularmente grandes y complejos. Pueden estar asociados a una cantidad de variables y restricciones que dificultan la resolución de los mismos de forma óptima. Por este motivo, es necesario utilizar técnicas de optimización matemática. Esto implica desarrollar un modelo de optimización definido en términos de variables de decisión, restricciones impuestas por los datos del problema y un algoritmo que resuelva la instancia del modelo.

Dentro de las técnicas más importantes de optimización matemática se destaca la Programación Matemática dentro de la cual se encuentra, por ejemplo, la Programación Lineal, Programación Entera, Programación Entera-Mixta y por otro lado la Programación con Restricciones.

La Programación con Restricciones es un paradigma de la programación en informática donde las relaciones entre las variables se expresan a través de restricciones [3]. Es una técnica utilizada para describir y resolver muchos problemas de la vida real como por ejemplo problemas de planificación y toma de decisiones entre otros.

Existen dos tipos de problemas de Programación con Restricciones, Problemas de Satisfacción de Restricciones y Problemas de Optimización de Restricciones. Un Problema de Satisfacción de Restricciones se define mediante un conjunto de variables asociadas a dominios que contienen los valores que pueden tomar y un conjunto de restricciones que restringen los valores que estas variables pueden tomar. La solución de un problema de este tipo estará determinada cuando exista un conjunto de valores asignados que satisfacen todas las restricciones.

En un problema de Optimización de Restricciones la solución no solamente deberá satisfacer todas las restricciones, sino que además estará afectada por una función objetivo que determinará la calidad de la solución.

La implementación de esta técnica requiere por un lado modelar el problema en términos de variables, dominios, restricciones. Luego, el problema quedará sujeto a su resolución. Para ello se utilizan diferentes técnicas para la eliminación de valores inconsistentes en los dominios de las variables, así como también la utilización de algoritmos de búsqueda y heurísticas de ordenación. En general, éstas se combinan para realizar una búsqueda más eficiente.

2.1 Conceptos

Un problema de Programación con Restricciones quedará determinado mediante [4]:

- Un conjunto de variables X $\{x_1;....; x_n\}$: estas variables quedan definidas por las características del problema y son las incógnitas que se intentarán resolver.
- $D = < D_1;....; D_n >$: donde cada elemento representa el dominio que contiene los valores posibles que se le pueden asignar a cada variable.
- R es el conjunto que define las restricciones: éstas restringen los valores que las variables pueden tomar. En el caso de las restricciones, existen tres tipos: unarias, binarias y no-binarias, considerando una, dos o muchas variables asociadas respectivamente.
  Ejemplos:
  - La restricción $x < 18$ es una restricción unaria sobre la variable x.
  - La restricción $x_1 + x_2 = 4$ es una restricción binaria.

Si éste además es un problema de optimización de restricciones, la solución estará asociada a una función objetivo:

- Una función objetivo es una expresión que relaciona variables del problema con el objetivo de obtener un valor que interesa optimizar.

La asignación de variables implica almacenar un valor en el espacio de memoria correspondiente a cada una de ellas. Cuando todas las variables son asignadas y estos valores satisfacen todas las restricciones, estas forman parte de una solución global. En caso de que esta asignación sea parcial, es decir, que se asignen valores a un conjunto de variables del problema y estas satisfagan todas las restricciones, se dice que la solución es parcial.

A continuación, se detallan las principales técnicas para procesar las restricciones.

- Algoritmos de búsqueda: éstos son un conjunto de instrucciones que exploran el espacio de soluciones con el objetivo de encontrar una solución cumpliendo con las condiciones del problema.
- Técnicas de consistencia: estas técnicas tienen como objetivo eliminar valores inconsistentes de los dominios de las variables para, de esta forma, reducir el espacio de búsqueda complementando a los algoritmos de búsqueda.
- Heurísticas de ordenación de variables: consiste en seleccionar el orden en el cual las variables van a ser estudiadas e instanciadas para, de esta manera, mejorar la eficiencia de resolución.

2.2 Algoritmos de Búsqueda

Los algoritmos de búsqueda son un conjunto de instrucciones que exploran el espacio de soluciones generadas por las posibles asignaciones de valores que pueden tomar las variables del problema con el objetivo de encontrar una solución en caso de que exista [44]. La combinación de estas asignaciones de valores genera un espacio de búsqueda denominado comúnmente árbol de búsqueda.
.
En la Figura 1 se puede observar un árbol de búsqueda generado por las posibles asignaciones.

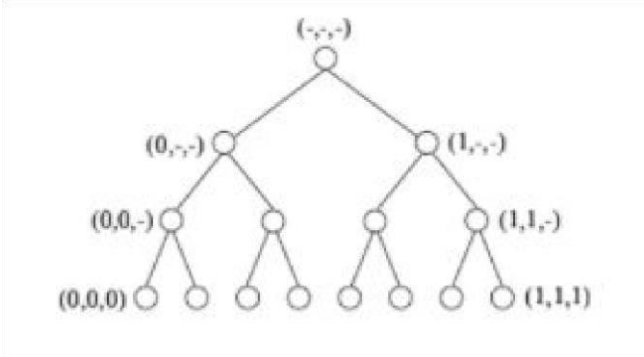

Figura 1: Árbol de Búsqueda, [44].

Este árbol de búsqueda queda definido por nodos intermedios y nodos hojas. Los nodos intermedios representan las asignaciones parciales de variables, es decir que no todas las variables están asignadas y los nodos hojas, de menor nivel, representan las posibles asignaciones de la totalidad de las variables representadas.

Los algoritmos de búsqueda se pueden clasificar en algoritmos completos e incompletos [4]. Los algoritmos completos exploran todo el árbol de búsqueda obteniendo una solución óptima o demostrando que el problema no tiene solución. Los algoritmos incompletos son más eficientes debido a que realizan la búsqueda en un espacio de solución acotado, es decir, que no exploran todo el árbol de búsqueda, pero por este motivo no garantizan que la solución encontrada sea óptima.

Se presentan a continuación los siguientes algoritmos sistemáticos: Generación y prueba, Backtracking y Chequeo hacia adelante.

2.2.1 Generación y Prueba

Este algoritmo recorre el árbol de búsqueda generado por el espacio de soluciones realizando todas las combinaciones de asignaciones completas [45]. Luego verifica si estas cumplen con las restricciones del problema, como se puede observar en la Figura 2. Por este motivo resulta ineficiente ya que a medida que realiza asignaciones parciales no verifica si estas pueden formar parte de la solución.

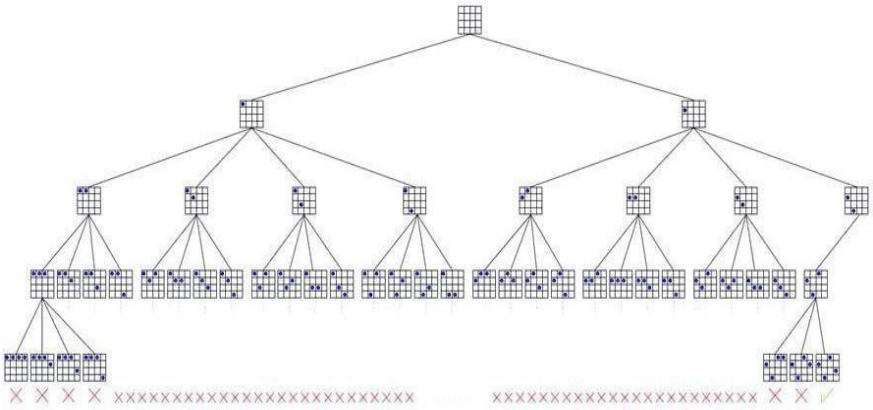

Figura 2: Ejemplo de búsqueda utilizando Generación y Prueba, [45].

2.2.2 Backtracking

El algoritmo Backtracking explora el árbol de búsqueda en profundidad, pero a medida que asigna un valor a la variable actual, verifica que sea consistente con la asignación parcial [4].

Si el valor de la variable actual es consistente con la asignación parcial, el algoritmo avanza en el árbol de búsqueda seleccionando una nueva variable realizando la misma verificación.

Si el valor de la variable actual no es consistente con la asignación parcial, el algoritmo retrocede y asigna un nuevo valor a la variable actual. Se repite este procedimiento hasta encontrar que forme parte de una solución parcial, avanzando nuevamente en el árbol de búsqueda o retrocediendo en caso de que agote el dominio que puede tomar la misma, como se puede observar en la Figura 3.

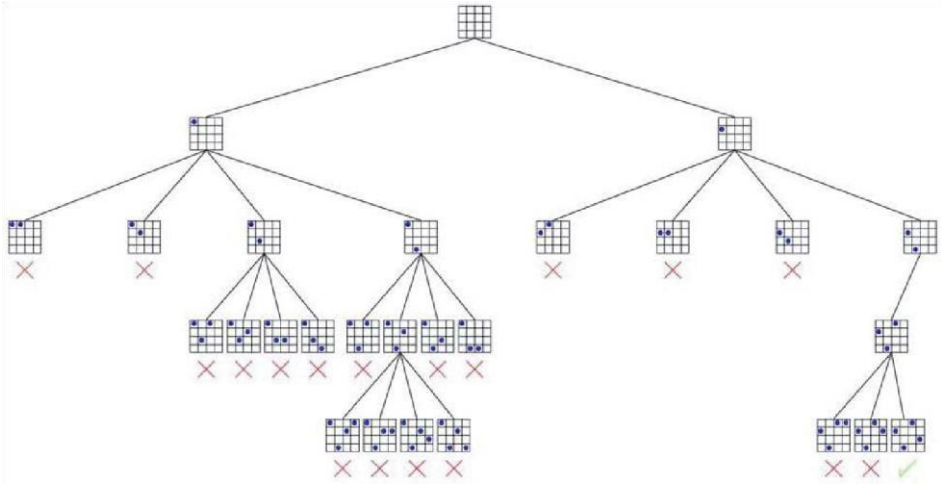

Figura 3: Ejemplo de búsqueda utilizando Backtracking, [45].

El algoritmo finaliza cuando todas las variables son consistentes y satisfacen todas las restricciones o cuando todas las combinaciones de los valores que pueden tomar las variables no conducen a una solución.

Si bien este algoritmo es similar al algoritmo de Generación y Prueba, resulta más eficiente que éste porque no sigue explorando en una rama de asignaciones parciales que no conduce a una solución [44]. De todas formas, es considerado ineficiente porque chequea la variable actual y las pasadas sin considerar si la variable actual es consistente con las futuras.

2.2.3 Chequeo hacia adelante

El algoritmo Chequeo hacia adelante, más conocido como Forward Checking, a diferencia de los algoritmos de Backtracking y Generación y prueba, realiza una verificación hacia adelante en cada una de las etapas de la búsqueda utilizando técnicas de consistencia.

De esta forma, verifica que la asignación actual sea consistente con los valores de las variables futuras, eliminando temporalmente los valores inconsistentes de los dominios de las variables futuras [44].

En el caso de que todos los posibles valores del dominio de una variable futura sean inconsistentes con la variable actual, se asigna un nuevo valor del dominio a la variable actual. Así es que, si ningún valor es consistente se realizará un backtrack explorando una nueva variable, como se observa en la Figura 4.

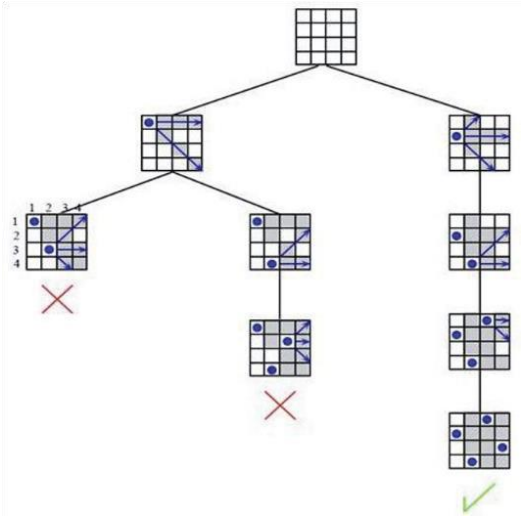

Figura 4: Ejemplo de búsqueda Chequeo hacia Adelante, [44].

2.3 Técnicas de Consistencia

Las técnicas de consistencia son utilizadas para detectar aquellos valores de los dominios de las variables que son inconsistentes para, de esta forma, reducir el espacio de búsqueda complementándose con los algoritmos de búsqueda logrando mayor eficiencia en términos de tiempos de resolución.

2.3.1 Consistencia de nodo

Este nivel de consistencia verifica todas las restricciones unarias sobre una variable eliminando del dominio de esta variable los valores que no cumplen con estas restricciones. De esta forma se logra reducir el espacio de búsqueda [44].

Ejemplo: La variable *x,* cuyo dominio es [0;110], y las restricciones unarias $x \geq 8$, $x \leq 99$. Los intervalos [0;7] y [100;110] son eliminados debido a que estos no conducirán a ninguna solución.

2.3.2 Consistencia de arco

Este nivel de consistencia considera las restricciones binarias. Es decir, que dadas dos variables x_i y x_j, para cada valor perteneciente al dominio de x_i verifica si existe algún valor del dominio de x_j que satisface las restricciones que relacionan x_i y x_j. De esta forma, los valores del dominio de x_i que no son consistentes con ningún valor del dominio de la variable x_j, son eliminados [45].

Ejemplo. Dadas dos variables x_i de dominio [2;5], x_j de dominio [9;11] y la restricción binaria $x_i \prec x_j$ se puede observar que todos los valores del dominio de x_i son consistentes.

2.4 Heurísticas de ordenación

De forma de lograr que los algoritmos de búsqueda sean más eficientes en términos de tiempo de resolución, es importante definir estrategias de enumeración seleccionando el orden en el cual las variables van a ser estudiadas e instanciadas [44].

Esto es posible lograrlo mediante la utilización de heurísticas para la ordenación de variables y valores.

Las heurísticas de ordenación de variables tienen como objetivo definir el orden en el que las variables van a ser asignadas. Se priorizan las variables más restringidas para, de esta forma, identificar y eliminar del árbol de búsqueda aquellas asignaciones que no conducen a una solución.

Las heurísticas para la ordenación de valores tienen como objetivo seleccionar el valor de la variable actual que es más probable que conduzca a una solución. Para ello se priorizan en la búsqueda los valores de la variable actual que generen menos inconsistencias a las asignaciones futuras.

3. Características del relevamiento

A continuación, se presenta el relevamiento y análisis de la literatura orientada a Problemas de Planificación aplicado en diversas actividades que tiene que ver con Asignación de Tareas, Asignación de Recursos, Planificación de calendarios entre otros. Se orientó la búsqueda en Programación con Restricciones para de esta forma poder determinar en qué etapa de desarrollo se encuentra esta área que resulta interesante investigar.

Como herramienta de búsqueda se utilizó Timbó [5], en particular las colecciones Science Direct y Springer. Éstas nos permitieron acceder a una amplia gama de trabajos que abordan problemáticas similares a nuestro campo de estudio.

Se llevó a cabo el registro de la literatura relevada en una planilla Excel, donde para cada uno de ellos se ingresaron entre otras cosas un breve resumen sobre el artículo, cuáles eran sus autores y colección o referencia de donde se obtuvo.

De forma de determinar la importancia de estos trabajos se priorizaron en orden de relevancia teniendo en cuenta las similitudes que tenían con nuestro caso de estudio y el contenido del mismo. Se relevaron dentro de los artículos encontrados las referencias bibliográficas considerando que en ellas se podrían encontrar trabajos relacionados.

Se consideró importante relevar también las herramientas utilizadas para resolver este tipo de problemas en los casos en los que se mencionan.

De forma de organizar los artículos relevados se clasifican en:

- Planificación de Eventos
- Planificación Académica
- Planificación en transporte y movilidad
    - Aerolíneas
    - Transporte de Carga
    - Autobuses/Trenes
    - Bicicletas
- Planificación en la Industria
- Otros

A continuación, se desarrollará el relevamiento realizado según la clasificación previamente mencionada.

3.1 Planificación de Eventos

Dentro de esta clasificación en la literatura relevada se observan diversos problemas como por ejemplo la programación de competiciones deportivas, organización de eventos sociales y culturales.

En el ámbito deportivo en [6], [7], y [8] se desarrolla la programación de torneos y competiciones deportivas de forma exitosa. En [7] Abderrahmane Aggoun y Alkis Vazacopoulos desarrollan un ejemplo de Programación de torneos en ligas deportivas que fue presentado en [6] pero enfocado en el sistema CHIP es un sistema de programación con restricciones, es resuelto mediante técnicas de Programación con Restricciones y presentan mejoras al algoritmo reduciendo el árbol de búsqueda.

En [7] se plantean las tendencias futuras como el desarrollo de aplicaciones híbridas, por ejemplo, combinación de Programación lineal de enteros mixtos y Programación con Restricciones, Metaheurísticas y Programación con Restricciones y el desarrollo de técnicas de optimización.

Un enfoque de programación de deportes en dos fases se desarrolla en [8], primero identificando una solución que designa a los participantes y programa cada una de las competiciones, luego asigna a cada competidor como el equipo local o visitante. Se realizan experimentos computacionales y los resultados se comparan con un enfoque de programación orientada a objetos. El enfoque de Programación con Restricciones logra soluciones óptimas para problemas con hasta dieciséis equipos y soluciones casi óptimas para problemas con hasta treinta equipos.

En el ámbito social, en [9] se desarrolla un problema de asignación, el problema surge en el contexto de la organización de un "partido progresista" en un encuentro de yates. Algunos yates son anfitriones y las tripulaciones de los yates restantes deben visitarlos considerando determinadas restricciones, se aborda el problema de dos formas distintas resultando que la diferencia en el rendimiento entre los dos enfoques Programación Lineal y Programación con Restricciones es particularmente marcado, no pudiendo encontrar una solución usando Programación Lineal. En problemas particularmente grandes como este el éxito del enfoque abordado con programación con restricciones se debe a la elección cuidadosa de heurísticas para dirigir su búsqueda.

En el ámbito cultural, los anuncios publicitarios en los espacios televisivos es un caso de estudio interesante abordado en [10], donde el departamento de marketing de un paquete de canales decide modificar su oferta comercial con el fin de vender paquetes de spots publicitarios. Se proponen varios paquetes, caracterizados por su precio, tamaño y otros factores como la franja horaria. Se estudian varias resoluciones mediante enfoques que incluyen programación lineal, relajación lagrangiana, programación con restricciones y búsqueda local. Dentro de la comparativa que realizan, es interesante destacar que la combinación de la programación con restricciones y la búsqueda local conduce a un algoritmo robusto y eficiente.

En [7], [8], [9], y [10] la Programación con Restricciones en este tipo de problemas resulta más eficiente en comparación con otros enfoques y a esto se agrega que los algoritmos de búsqueda mejoran la eficiencia de estos modelos.

3.2 Planificación Académica

En el ámbito académico, uno de los problemas de planificación más difíciles de resolver, dada la cantidad de restricciones a considerar y los imprevistos que pueden surgir, es la planificación de horarios y exámenes. Para ello, no solo se necesita contar con herramientas que resuelvan de forma eficiente este tipo de problemas, sino que también puedan recalcular la planificación a medida que surjan imprevistos considerando los mismos como parte de la naturaleza de este tipo de problemas como la ausencia de docente y la disponibilidad de recursos en un momento dado, entre otros.

Dentro de la planificación de horarios de clases en universidades en [11], [7] se aborda esta problemática resolviendo un cronograma de 18 semanas y uno semanal respectivamente. En [11] se trata el problema de planificación de horarios de clase en una universidad, el cual es básicamente un problema de asignación de recursos, los cuales comúnmente se clasifican como NP-completos. En este tipo de problemas puede resultar difícil encontrar una solución óptima dado el enorme espacio de soluciones que presentan. Vale aclarar que el caso que se trata en este artículo es sumamente dinámico, los autores explican que, debido al rápido desarrollo económico y el rápido crecimiento de la población, las condiciones académicas se actualizan con frecuencia. Esto indicaba que el modelo a desarrollar debía ser flexible y adaptable, de forma que admitiera cambios. El objetivo de este trabajo se centró en encontrar un horario factible. Se resuelve un problema de datos reales, que consta de un calendario de 18 semanas, 1673 horas de clase, y 21 profesores, el cual se modela con Programación con Restricciones. El mismo se resuelve en un tiempo menor a 33 minutos, cuando en caso de resolverse manualmente, se tardan varias semanas. Es importante mencionar que se logró que el modelo puede modificarse fácilmente, adaptándose así al cambio. Los autores finalizan el proyecto sugiriendo que futuras investigaciones podrían incluir técnicas de relajación, lo cual es importante para resolver problemas altamente restringidos.

En el ámbito universitario en [7] se destaca la Programación con Restricciones como método de resolución para problemas de planificación. Enumera ejemplos en diferentes áreas, en particular, planificación de horarios en universidades. Los autores comienzan con una breve descripción del sistema CHIP, un sistema de Programación con Restricciones diseñado para abordar problemas de búsqueda restringida. Luego describe un conjunto de aplicaciones exitosas en las áreas mencionadas anteriormente. Entre algunas de los modelos en los que es aplicable la herramienta, se encuentra el ejemplo del horario universitario donde el director de una universidad está a cargo de establecer el horario semanal para dos clases. El objetivo es hacer que un horario esté sujeto a varias restricciones, dentro de las cuales se encuentran que un profesor no dicte dos clases a un mismo grupo en un mismo día, que cada profesor dicte una sola materia y restricciones relacionadas con los horarios que los profesores tienen disponible para dictar clase en la institución. Ambos grupos tienen los mismos cursos, todos con una duración de dos horas con los mismos profesores, exceptuando las clases de deportes y matemáticas. El modelo devuelve numerosas soluciones de las cuales los autores muestran dos y destacan que este tipo de problemas son fáciles de resolver mediante Programación con Restricciones.

En cuanto a problemas de horarios en liceos, en [12] se presenta la situación en donde los profesores dan clase a varios grupos mientras que los estudiantes permanecen en los salones. Además de un enfoque de Programación con Restricciones, también se utilizan modelos de Investigación de Operaciones, y técnicas de Búsqueda Local para ayudar al proceso de búsqueda de Programación con Restricciones al reducir efectivamente el espacio de búsqueda

de la solución. Se utilizaron modelos relajados que se pueden resolver utilizando algoritmos de coincidencia de costo mínimo para calcular los límites inferiores del problema en varias instancias del proceso de solución. Estos límites se utilizaron a su vez para priorizar las opciones de búsqueda del proceso de Programación con Restricciones. El uso del modelo de coincidencia de costo mínimo en el proceso de búsqueda es un mecanismo económico y eficiente para la creación de estrategias de búsqueda efectivas y es una forma competitiva de introducir información de dominio problemático en el entorno de Programación con Restricciones.

3.3 Planificación en transporte y movilidad

Se consideran diversos artículos en esta área aplicados a transporte de carga, autobuses, trenes, aerolíneas y bicicletas compartidas. Los mismos intentan resolver problemas de gestión logística.

3.3.1 Aerolíneas

En el rubro aéreo se relevaron artículos en los cuales se intentan resolver problemas de gestión de la operación en este sentido en [13], [14], [15] se resuelven tres problemas de asignación: de tripulación, de puertas en aeropuertos y de tramos de vuelo respectivamente.

El problema de asignación de la tripulación de la aerolínea consiste en asignar líneas de trabajo a un conjunto de miembros de la tripulación de manera que se divida un conjunto de actividades y se minimicen los costos de esa asignación. Este problema es abordado en [13], particularmente para las compañías aéreas europeas. En este trabajo se desarrollan dos algoritmos diferentes para abordar el problema de la asignación de tripulaciones de aerolíneas.

El primero es una aplicación del marco de generación de columnas basado en la Programación con Restricciones, el segundo enfoque realiza una búsqueda de árbol heurística basada en Programación con Restricciones. Presentan cómo se pueden acoplar ambos algoritmos para superar sus debilidades inherentes mediante la integración de métodos de la Programación con Restricciones y la Investigación de Operaciones. Los resultados numéricos muestran la superioridad del algoritmo híbrido en comparación con la búsqueda de árbol basada en Programación con Restricciones y la Generación de Columnas solamente.

La asignación de puertas en aeropuertos para vuelos entrantes y salientes con el aumento de la intensidad del tráfico se ha vuelto más importante y complicada: en [14] se aborda esta problemática. La asignación incorrecta de puertas para vuelos entrantes y salientes puede resultar en retrasos, insatisfacción de los clientes y aumento de los costos operativos. Se proponen dos modelos para resolverlo: uno de Programación Entera y un modelo de Programación con Restricciones. El primero es capaz de encontrar la solución óptima en unos 100 segundos, mientras que el segundo termina después de 1800 segundos con una solución casi óptima. Se destaca que esto no implica que el modelo de Programación Entera sea superior al modelo de Programación con Restricciones, sino que debe revisarse en términos de los métodos de búsqueda. Para obtener más direcciones de investigación, continúan realizando estudios y plantean una línea interesante de abordar cómo utilizar en conjunto ambos modelos, modelo híbrido y además evaluar heurísticas de búsqueda en formulaciones de programación con restricciones.

Dentro del área de la planificación operativa de la aerolínea a corto plazo, la asignación de cola es el problema de asignar tramos de vuelo a aeronaves identificadas individualmente mientras se satisfacen todas las restricciones operativas y se optimiza alguna función objetivo. En [15] se presenta un enfoque de solución de programación con restricciones y generación de columnas híbridas. Este enfoque se puede utilizar para producir rápidamente soluciones para la gestión de operaciones y también para producir soluciones casi óptimas para escenarios de planificación a largo y medio plazo.

3.3.2 Transporte de Carga

Uno de los desafíos más importantes en el transporte de carga es obtener ahorros significativos con el objetivo de maximizar las ganancias y a la vez ser más competitivos. Dentro de los factores que se buscan optimizar, se pretende hacer una adecuada asignación de choferes y minimizar el costo de actividades no operativas como por ejemplo tiempos de espera, distancias que se recorren sin carga.

El problema del transporte de la cosecha de caña de azúcar en Australia se trata en [16]. Disminuir de forma eficiente los costos del mismo puede generar ahorros sustanciales dado que el transporte significa el 35% de los costos totales de manufactura. Anteriormente, en [17], se había desarrollado un modelo matemático basado en heurísticas para resolver este problema, requiriendo una entrada mínima del usuario dando buenos resultados para grandes redes en poco tiempo. Sin embargo, si ocurre algún evento inesperado, el algoritmo no puede modificar el horario existente, sino crear un programa totalmente nuevo. La programación con restricciones es flexible y permite absorber modificaciones en caso de ser necesario. Se concluye que esta herramienta es aplicable al problema, y no sólo permite la reprogramación de una forma ágil, si no que disminuye los costos de desarrollo del sistema de programación.

En la industria forestal en [18] los autores presentan el problema de la programación de transporte de troncos, este trabajo propone un método de solución basado en Programación con Restricciones y Programación Matemática. El problema consiste en programar el transporte de troncos entre áreas forestales y aserraderos, así como enrutar la flota de vehículos utilizados para tal fin. El objetivo consiste en minimizar el costo total de actividades no productivas tales como el tiempo de espera de camiones, los operarios que cargan los mismos y la distancia que recorren los vehículos vacíos. Se utiliza un modelo de Programación con Restricciones para abordar el problema combinado de programación y enrutamiento y Programación de Enteros para hacer frente a la optimización de los tiempos muertos. Se toman dos casos de estudio brindados por FERIC, (Forest Engineering Research Institute of Canada), resolviéndolos a su vez con dos enfoques. Un modelo específico de Programación con Restricciones y un modelo híbrido, que combina Programación con Restricciones y Programación Entera. En varios casos el enfoque híbrido proporciona una mejor solución que el enfoque de Programación con Restricciones. Sin embargo, para algunos casos, se incrementa el tiempo de espera de los operarios.

3.3.3 Autobuses/Trenes

En [19] se desarrolla una aplicación para optimizar las tareas del conductor del autobús donde el algoritmo de Generación de Columnas se usó para combinar la Programación con Restricciones y Programación Lineal.

Esta aplicación genera un servicio de autobús por un solo día con el objetivo de asignar a los conductores determinadas secuencias de viajes. Todas las decisiones relativas a los viajes en autobús o el horario, ubicación de inicio, ubicación de finalización, hora de inicio y hora de finalización se han realizado antes de esta etapa. Este documento muestra que las dos tecnologías se pueden combinar para extraer los puntos fuertes de ambas. Destacan nuevamente la eficiencia de la Programación con Restricciones y algoritmos híbridos.

En [20] se aborda el problema de la asignación de la tripulación en el transporte, al tiempo que se satisface una variedad de regulaciones y reglas laborales sindicales. El objetivo que se busca al resolver este problema es asignar la tripulación al medio de transporte, minimizando el costo de operación de ésta, mejorando la calidad de vida de la tripulación y cumpliendo con todas las regulaciones. Los estudios existentes sobre este problema se han enfocado en diferentes tipos de transporte, como ser ómnibus, trenes y aerolíneas. E n este artículo, el estudio se centra en un sistema de tránsito masivo rápido (MRT) de Taipei, el cual se resuelve a través de Programación con Restricciones, a diferencia de los estudios realizados hasta ese momento en los cuales este problema sólo se había resuelto utilizando métodos heurísticos y metaheurísticos. La resolución consta de dos fases la generación y la optimización de tareas, este último se basa en un modelo de problema de cobertura de conjuntos. Los resultados obtenidos lograron cumplir con todas las reglas de planificación utilizadas por el MRT de Taipei, siendo que la solución manual no logró cumplir con los requisitos de descanso. También se concluyó que se requieren 55 tareas de la tripulación en vez de 58 que eran las definidas de forma manual por una persona con experiencia del MRT reflejándose una disminución en los costos del 5,2%, además según los resultados obtenidos, se puede disminuir también los turnos de trabajo de 14 a 11 turnos. Es importante resaltar que se obtiene una solución en 307 segundos, mientras que manualmente esta tarea tomaba alrededor de una semana.

3.3.4 Bicicletas

Los sistemas de bicicletas compartidas, ponen a disposición de un grupo de usuarios una serie de bicicletas para que sean utilizadas temporalmente como medio de transporte. Permiten recoger una bicicleta y devolverla en un punto diferente. El problema que se plantea en estos casos es balancear los puntos donde se localizan las mismas para evitar la falta o exceso de bicicletas respecto a la demanda.

En [21] se trata sobre los sistemas de bicicletas compartidas. Este artículo continúa el estudio previamente realizado por los autores en [22] y [23]. Las estaciones de bicicletas siguen patrones con respecto a cuándo están vacías o llenas. Es de esperar que, en la hora pico de la mañana, las estaciones que se encuentran cerca de los centros de las ciudades se encuentren llenas, mientras que las estaciones más cercanas a la periferia se encuentren vacías. Este ejemplo ilustra dos problemáticas: algunos usuarios no van a encontrar bicicletas en las estaciones cercanas, mientras que otros van a encontrar estaciones llenas al momento de devolverlas. Ambas situaciones llevan a los usuarios a sentirse insatisfechos, optando así por dejar de usar el servicio. Para evitar esto, este tipo de sistemas toma medidas para equilibrar la cantidad de bicicletas en las distintas estaciones. Lo que se hace usualmente en estos casos es redistribuir las bicicletas en la noche con una flota de camiones, la dificultad que esto presenta es que en general la cantidad de estaciones de bicicletas es mucho mayor que la cantidad de camiones, lo que deriva en tener que elegir qué estaciones redistribuir, y posteriormente, definir el ruteo de los camiones para poder distribuir las bicicletas en un tiempo razonable. La solución se realiza en dos modelos diferentes, ambos utilizando programación con restricciones: primero, un modelo de enrutamiento basado en la formulación clásica del problema de enrutamiento del vehículo (VRP - Vehicle routing problem) y un modelo de pasos que encuentra una ruta para de “K” pasos para cada vehículo, donde el primero y el último son el depósito de bicicletas, resultando así que cada vehículo realiza K-1 paradas. Se evalúa el desempeño de ambos modelos en una evaluación experimental. Luego, se utilizan los dos modelos de Programación con Restricciones como base para una estrategia de búsqueda de Large Neighborhood Search, que es personalizado para explotar las características del modelo de Programación por Restricciones. El problema se resuelve en poco tiempo de ejecución y se concluye que el modelo es competitivo con lo relevado previamente en el estado del arte. La Programación con Restricciones le da una flexibilidad importante al problema, ya que alcanza con agregar o eliminar restricciones. Esto es sumamente interesante dado que cada problema de reequilibrio de bicicletas compartidas tiene su propia formulación y no existe una genérica para todos, por lo que requiere una formulación flexible.

3.4 Planificación en la Industria

Dentro del sector industrial, se encuentran problemas de planificación tanto de producción como de tareas de mantenimiento de máquinas en plantas industriales.

Cuando se habla de planificación industrial nos referimos a organizar los recursos de manera tal para que sean eficientes, efectivos y eficaces. Para ello, no sólo se debe organizar los recursos con los que se cuenta sino también prever una gran cantidad de variables que pueden cambiar al interior de una planta.

La planificación de la producción es abordada en [24]. La investigación pone el foco en la solución de problemas de satisfacción de restricciones dinámicos, si bien la mayoría de los algoritmos de satisfacción existentes son para problemas estáticos, en las aplicaciones de la vida real no resulta llamativo que las restricciones evolucionan debido al entorno o a las interacciones del usuario. Los autores concluyen que a partir de la aplicación del sistema CHIP en tres industrias, la programación de producción basada en restricciones no solo sirve como una herramienta de programación simple, sino que también proporciona una base para la toma de decisiones para la gestión de empresas. Con las amplias aplicaciones del modo de fabricación bajo demanda, el uso de programación con restricciones en los sistemas APS, (Advanced Planning and Scheduling; son sistemas que planifican la producción y el stock de acuerdo a recursos disponibles y capacidad de planta), ha aumentado considerablemente.

La industria automotriz produce cientos de pruebas en prototipos de automóviles previamente a comenzar la producción en masa. Dado que los prototipos son manufacturados, éstos son muy caros. Sin embargo, un mismo prototipo puede ser utilizado para varias pruebas. En [25] se aborda el problema de programar todas las pruebas minimizando la cantidad de prototipos a realizar, lo que se traduce en minimizar los costos de producción de prototipos. Se debe tener en cuenta también que existen cientos de variantes de prototipos dependiendo de la combinación de varios componentes; de esta forma cada prueba se asigna al prototipo correspondiente en concordancia con los requerimientos de sus componentes. Para resolver el problema, los autores utilizan un modelo de Programación con Restricciones que minimiza el número de prototipos necesarios. Sin embargo, aunque este enfoque ayuda a obtener una solución factible, aún hace falta encontrar un buen límite inferior para la medición de la brecha de optimización dado que Programación con Restricciones no es un método adecuado en esta situación, ya que puede ajustar el límite inferior solo para los nodos explorados localmente en la misma rama del árbol de búsqueda. Por el contrario, la Programación Entera proporciona una perspectiva global ya que el proceso de bifurcación y límite siempre eleva el límite inferior. Por lo tanto, los autores sugieren un modelo de programación entera simplificado que se puede resolver para una instancia grande y proporcionar una solución considerada como un límite inferior para este problema. Se aplican ambos enfoques en cuatro escenarios, que van desde 40 a casi 500 pruebas. El resultado es que, para problemas pequeños, el enfoque de Programación con Restricciones encuentra una solución óptima, pero cuando el problema crece, sólo se encuentran soluciones factibles. Con el modelo de Programación Entera simplificado, se obtienen los valores de límite inferior del número de prototipos requeridos. Los resultados muestran que las brechas entre las soluciones obtenidas de los dos modelos son sólo un prototipo como máximo. Esto significa que no solo el enfoque de Programación con Restricciones puede lograr buenas soluciones viables, sino que también el modelo de programación entera simplificado puede proporcionar límites más bajos.
[26] aborda el problema de la planificación de la producción y las tareas de mantenimiento preventivo de la planta. El caso de estudio consiste en una planta de tipo Job Shop flexible, (FJS). Este problema ha sido abordado con distintas metodologías: algoritmos combinatorios,

heurísticas, metaheurísticas, colonia de hormigas, entre otros. Uno de los problemas que presentan estas metodologías es la escasa capacidad que tienen para incorporar características adicionales propias de los ambientes industriales. La Programación con Restricciones permite esto siendo mucho más flexible. Las plantas de tipo FJS tienen un conjunto de máquinas que llevan a cabo diferentes tareas, pero sólo pueden ejecutar una operación a la vez. Con el objetivo de modelar este problema, se considera que en el período de planificación de las tareas, se incluyen las de mantenimiento preventivo, con lo cual, la planificación incluye asignar máquinas a las operaciones que requiera cada Job Shop, definir el orden de las tareas asignadas a cada máquina, especificar el tiempo de inicio de cada operación y de cada tarea de mantenimiento. Además, se hicieron suposiciones como que se conoce de antemano la duración de cada actividad de mantenimiento, estas actividades no pueden ser interrumpidas y las máquinas no sufren desperfectos durante el tiempo de planificación, entre otros. El modelo se validó con varios ejemplos tomados de la literatura, así como de otros creados a partir de los anteriores. Se encontraron soluciones óptimas en bajos tiempos de ejecución, encontrándose soluciones de igual o mejor calidad que las reportadas en la bibliografía del artículo.

En [27] se presenta un problema de planificación de la producción real en donde hay muchos más productos que máquinas de la industria química, se resuelve un problema de planificación y programación con programación combinada de Enteros Mixtos y Restricciones. El algoritmo combinado MIP/CP, es capaz de encontrar una buena solución factible a un problema que de otro modo sería intratable en términos de tiempo. Se destaca que la clave de la eficiencia del sistema híbrido fue la cuidadosa asignación de aspectos de los dos solucionadores.

Dentro de la programación y planificación de las cadenas de suministro, en [28] se aborda un problema de este tipo. Las tareas deben asignarse a las instalaciones y asignaturas programadas para publicar fechas y plazos. Las tareas pueden ejecutarse en paralelo en una instalación determinada siempre que el consumo total de recursos en cualquier momento permanece con límites, (programación acumulativa). El problema se descompone naturalmente en una parte de asignación y una programación, aprovechando de esta forma las fortalezas relativas de la Programación Lineal Entera Mixta (MILP) y Programación con Restricciones aplicando MILP al problema de asignación y Programación con Restricciones al problema de la programación. Luego vinculan los dos con un algoritmo de Benders basado en lógica obteniendo aceleraciones de varios órdenes de magnitud en relación con el estado del arte existente tanto en la Programación de Enteros Mixtos, (CPLEX), como en la Programación con Restricciones, (ILOG). Como resultado, resuelven instancias más grandes a la optimización de lo que se podría resolver previamente. En problemas mínimos de makespan, el método Benders proporciona una solución y un límite inferior en la amplitud óptima incluso cuando se termina antes encontrar una solución demostrablemente óptima.

Emrah B. Edis presenta en [29] el problema de balanceo de línea de desmontaje con decisiones de secuenciación, (Disassembly Line Balancing and Sequencing problem, o DLBS). En la recuperación de productos, el desmontaje, (desarmar un producto en sus partes constituyentes), es una operación significativa. Las operaciones de desmontaje son generalmente realizadas en una línea que consta de una serie de estaciones de trabajo. Al momento de distribuir las tareas entre las estaciones de trabajo en la línea, los principales problemas son los siguientes: minimizar el número de estaciones de trabajo que también minimicen el tiempo total de inactividad, lograr una distribución equilibrada de tareas entre las estaciones de trabajo, (es decir, balanceo de la línea de desmontaje), y la potencial peligrosidad de las piezas; si las hay, deben retirarse lo antes posible para evitar una posible explosión. Otro problema es la demanda de piezas para desmontar: algunas piezas pueden resultar dañadas en el proceso y pueden no ser aceptadas

por el cliente. Por lo tanto, las partes con demandas relativamente altas deben ser extraídas lo antes posible para reducir la escasez en la satisfacción de la demanda asociada. Al retirar las piezas en una línea de desmontaje, también puede ocurrir un requisito de cambio de dirección del producto. Por tanto, se pretende minimizar el número total de cambios de dirección a lo largo del proceso de desmontaje. Los dos primeros problemas están relacionados con la asignación de tareas a las estaciones de trabajo, mientras que los últimos tres refieren a la secuencia de tareas a lo largo de la línea de desmontaje, además de las decisiones de asignación. Hasta ahora, para resolver este problema, se han aplicado diversos enfoques incluidos procedimientos heurísticos, enfoques metaheurísticos y enfoques exactos. Para este problema, se desarrolla un modelo de CP genérico con fases de búsqueda integradas y se evalúa su desempeño mediante una serie de problemas de desmontaje e instancias de referencia de tamaño pequeño, mediano y grande. Las principales contribuciones de este estudio son dos: en primer lugar, el autor afirma que es la primera vez que se utiliza el enfoque de Programación con Restricciones aplicado al problema DLBS y por otro lado, los enfoques propuestos han mejorado significativamente las mejores soluciones, (encontradas hasta el momento), o han establecido nuevas soluciones de referencia para numerosos problemas de desmontaje. El modelo propuesto se puede utilizar para problemas de DLBS con hasta 35 tareas de desmontaje mientras que el procedimiento de solución propuesto se puede utilizar para encontrar soluciones rápidas y eficientes para problemas DLBS de gran tamaño.

Novara y Henning abordan en [30] el problema de la planificación reactiva en una planta con procesos batch multiproducto. La planificación reactiva, (“scheduling reactivo” o “rescheduling”), consiste en la revisión, adaptación y corrección de un “schedule” activo, debido a la ocurrencia de eventos imprevistos; se trata de encontrar encontrar la mejor manera de reaccionar a imprevistos luego de que los mismos ocurran. Entre los posibles imprevistos se cuentan: ingreso de una nueva orden de producción, falla de la maquinaria, falla en una unidad, (el batch que estaba siendo procesado), entre otros. Los autores proponen un modelo basado en programación con restricciones, dado que emplea un novedoso enfoque multi-objetivo que apunta a preservar la estabilidad de la planificación original, minimizando así el número de cambios que sobre ella se efectúan, a la vez que se trata de lograr un buen desempeño con relación a la medida de performance con la que se generó el “schedule” vigente. Se define un sistema de penalizaciones como forma de cuantificar la calidad de la solución obtenida a partir del modelo. Si bien el mismo se validó en diversos ejemplos basados en un caso de estudio de “scheduling” predictivo abordado por otros autores, los autores describen los resultados obtenidos en uno solo. Se simula la ruptura de una máquina, mostrando el período de tiempo en el cual no estará disponible, afectando así a cuatro batches. El resultado obtenido es muy bueno teniendo en cuenta que la mayoría de las modificaciones tienen lugar fuera del período crítico, sólo dos tareas son reasignadas a otras unidades y las tareas que se atrasan, (aunque son muchas), no lo hacen en el período crítico.

3.5 Otros

Se relevaron diversos artículos en los cuales se destaca la potencialidad de la Programación con Restricciones y se indican las áreas donde esta técnica se destaca.

En [31], [32] se destaca la Programación con Restricciones como una técnica natural, que ofrece facilidades para modelar problemas y la presentan como una plataforma flexible para resolverlos. Los autores destacan en [31] el crecimiento en el uso de esta técnica. Se mencionan diversas áreas donde ha sido aplicada exitosamente, por ejemplo, áreas de programación, transporte, asignación de personal y asignación de recursos, cronogramas dinámicos como en [33]. En [34] los autores demuestran experimentalmente que para un conjunto de problemas de asignación la Programación con Restricciones se desempeñó consistentemente mejor que la Programación Entera con la cual obtienen un rendimiento variable e impredecible. Para ambas técnicas, el rendimiento se rige por la capacidad de reducir el espacio de búsqueda.

En [7] los autores destacan la Programación con Restricciones como un método de resolución para problemas de planificación en general, enumera ejemplos en diferentes áreas donde se ha aplicado exitosamente como deportes, producción industrial y planificación de horarios en universidades.

Es de destacar la importancia de la Programación con Restricciones para explotar otras técnicas utilizando en donde se combina la Programación con Restricciones con Programación lineal, Técnicas de Investigación de Operaciones como en [33] para resolver problemas de cronograma, enteros mixtos, metaheurísticas, logrando buenos resultados en búsqueda de una solución factible en problemas donde de otro modo no resultaría tan eficiente o incluso no se logra obtener una solución. En este sentido en [32] describen cómo se puede utilizar para explotar la programación lineal dentro de diferentes tipos de algoritmos y cuáles son sus beneficios. En particular, puede potenciar técnicas como la Relajación Lagrangiana, Descomposición de Benders y Generación de Columnas. Concluye que el mayor beneficio de Programación con Restricciones es el soporte que ofrece para la encapsulación de modelos y reutilización.

En [7] comienzan con una breve descripción del sistema CHIP que es un sistema de Programación con Restricciones diseñado para abordar problemas de "búsqueda restringida". Luego describe un conjunto de aplicaciones exitosas en las áreas de planificación de la producción y programación. Finalmente, detalla los modelos de Programación con Restricciones de tres ejemplos de horarios y programación deportiva. Dentro de las aplicaciones exitosas se destacan: Una herramienta de configuración en la industria de las telecomunicaciones, la programación de producción continua en industrias químicas problema de programación que requiere un enfoque híbrido que combina Programación de Enteros Mixtos, (MIP), y Programación con Restricciones, EVA: que es una aplicación desarrollada para EDF, (Electricidad de Francia), para planificar el transporte de combustible nuclear desperdiciado, OPTI-CHANNEL de COSYTEC es una planificación herramienta para recursos y actividades en el negocio audiovisual. Otro ejemplo de programación de la tripulación es la herramienta GYMNASTE, que crea un horario de turnos de enfermeras sujeto a varias restricciones complejas.

Dentro de los ejemplos de horarios y programación deportiva: el ejemplo del horario universitario donde el director de una universidad está a cargo de establecer el horario semanal para dos clases en el último año de la universidad. El objetivo es hacer que un horario esté sujeto a varias

restricciones. El ejemplo de la fiesta progresiva: el problema es organizar una fiesta para 42 equipos de yates en una regata. Programación de la liga deportiva: el problema se refiere a la programación de torneos mentores en ligas deportivas como fútbol, hockey y baloncesto. Deja planteado a futuro modelos híbridos por ejemplo de Programación con Restricciones y Metaheurísticas o Programación de Enteros Mixtos con Programación con Restricciones.

También en la línea de las universidades en [7] se destaca la Programación con Restricciones como método de resolución para problemas de planificación. Enumera ejemplos en diferentes áreas: deportes, producción industrial y planificación de horarios en universidades. Los autores comienzan con una breve descripción del sistema CHIP: un sistema de Programación con Restricciones diseñado para abordar problemas de "búsqueda restringida". Luego describe un conjunto de aplicaciones exitosas en las áreas mencionadas anteriormente. Entre algunas de los modelos en los que es aplicable la herramienta, se encuentra el ejemplo del horario universitario donde el director de una universidad está a cargo de establecer el horario semanal para dos clases en el último año de la universidad, el objetivo es hacer que un horario esté sujeto a varias restricciones, dentro de las cuales se encuentran que un profesor no dicte dos clases a un mismo grupo en un mismo día, que cada profesor dicte una sola materia y restricciones relacionadas con los horarios que los profesores tienen disponible para dictar clase en la institución. Ambos grupos tienen los mismos cursos, (todos con una duración de dos horas), con los mismos profesores, exceptuando las clases de deportes y matemáticas. Los autores entienden que este problema se puede entender a través de un paralelismo con un problema de producción: los profesores serían análogos a las máquinas mientras que las clases serían análogas a las operaciones. El modelo devuelve numerosas soluciones de las cuales los autores muestran dos, y destacan que este tipo de problemas son fáciles de resolver con programación con restricciones.

El ejemplo de la fiesta progresiva: el problema es organizar una fiesta para 42 equipos de yates en una regata. Programación de la liga deportiva: el problema se refiere a la programación de torneos mentores en ligas deportivas como fútbol, hockey y baloncesto. Deja planteado a futuro modelos híbridos por ejemplo de Programación con Restricciones y Metaheurísticas o Programación de Enteros Mixtos con Programación con Restricciones. El capítulo también detalla los modelos Programación con Restricciones de tres ejemplos de horarios y programación deportiva. Dentro de las aplicaciones exitosas se destacan: Una herramienta de configuración en la industria de las telecomunicaciones, la programación de producción continua en industrias químicas problema de programación que requiere un enfoque híbrido que combina Programación de Enteros Mixtos, (MIP), y Programación con Restricciones, EVA: que es una aplicación desarrollada para EDF, (Electricidad de Francia), para planificar el transporte de combustible nuclear desperdiciado, OPTI-CHANNEL de COSYTEC es una planificación herramienta para recursos y actividades en el negocio audiovisual. Otro ejemplo de programación de la tripulación es la herramienta GYMNASTE, que crea un horario de turnos de enfermeras sujeto a varias restricciones complejas.

En el rubro de la salud, resulta sumamente interesante el artículo [35], el cual presenta el caso de un hospital: ¿cómo hacer una programación diaria de las salas de operaciones? La complejidad que este problema conlleva, además de las mismas limitaciones de recursos que presentan los demás problemas de planificación, (como ser la cantidad de salas, tamaño de las mismas, recursos humanos disponibles, horarios disponibles de los cirujanos, así como del resto del personal presente en cada operación, instrumentos necesarios para cada operación, entre otros), es la urgencia con la que hay que tratar a ciertos pacientes, la cual es una restricción propia de la salud, y, lo que no es menor, la prioridad entre pacientes. El objetivo de este estudio

es ayudar al coordinador de quirófanos, (quien se encarga tanto de las salas de operaciones, así como de la sala de recuperación post anestésica), a programar las operaciones en las diferentes salas, así como asignar los recursos necesarios a cada block quirúrgico disponible. Es interesante resaltar que, de acuerdo con [36], muy pocos estudios toman en cuenta tanto las restricciones materiales, así como las de personal: en general, las restricciones que se contemplan en cuanto a los recursos humanos son la cantidad y disponibilidad de cirujanos, camilleros y anestesistas y algunas veces también de enfermeros y asistentes. Este artículo tiene en cuenta todas las restricciones mencionadas anteriormente, así como también la preferencia de los cirujanos, anestesistas y enfermeros y la afinidad entre dos personas para trabajar en equipo. Se modeló el problema de dos maneras: utilizando Programación Entera Mixta y Programación con Restricciones. Éstos se compararon para determinar cuál se enfrenta mejor con un problema altamente restringido. Se obtuvieron soluciones viables para la mayoría de los escenarios. El modelo de Programación Entera Mixta, diseñado para un problema muy limitado de la vida real, puede resolverse de manera óptima en un corto tiempo de cálculo. Este modelo no permite restricciones muy sofisticadas que se encuentran en la vida real sin que se vuelva difícil de construir y de resolver. Por otro lado, el modelo de Programación con Restricciones proporciona una formulación clara y más intuitiva que permite una mejor comprensión. Además, tiene la ventaja de que se pueden incluir restricciones no lineales y, en particular, se encuentra que cuantas más restricciones se agreguen, mejor funcionará el modelo ya que el tamaño del problema disminuye, al igual que el tiempo de cálculo requerido para encontrar soluciones factibles. Si bien el modelo de Programación con Restricciones suele ser más lento que el modelo de Programación Entera Mixta para encontrar soluciones óptimas, es interesante ver que es más eficiente en la generación de soluciones viables, especialmente para problemas altamente restringidos y frecuentes.

3.6 Relevamiento de herramientas informáticas

Hoy día, debido a la globalización, estandarización y el desarrollo de nuevas tecnologías, el mercado cuenta con diversas herramientas de software para resolver problemas en los cuales es aplicable la Programación con Restricciones como método de optimización de recursos: para lenguajes como C++, ILOG Solver y Gecode, por ejemplo; para Java, Koalog y Choco, y para Prolog, Eclipe o Sicstus, entre otros. A pesar de existir múltiples alternativas, se encontró que en los trabajos relevados se utiliza de forma predominante ILOG Solver. Los únicos dos trabajos que no utilizan este software son [7] y [15] en donde se utiliza el sistema CHIP y Visual Studio .NET, respectivamente.

En [37], Fernández y Hill hacen un estudio comparativo de 8 herramientas para resolver problemas de programación con restricciones: ILOG Solver, clp (FD), Oz, SICStus, IF/Prolog, B-Prolog, ECLiPSe y CHR. Los autores utilizan los sistemas para resolver diversos problemas y de esta forma poder hacer una correcta comparación. Se concluye que ILOG Solver es el sistema con los mejores resultados; es el más robusto: logra resolver un problema que consta de 1600 variables. El sistema clp (FD) si bien da buenos resultados, es más lento que ILOG Solver. Además, cuando el tamaño del problema se incrementa, el programa devuelve mensajes de error, indicando así que no logra adecuarse a medida que crece el número de variables. Por otro lado, el sistema Oz resultó más rápido que ECLiPSe y casi tan rápido como clp, sin embargo, falló en uno de los problemas a los que se lo sometió. SICStus e IF/Prolog tuvieron resultados similares, siendo de dos a tres veces más rápido que ECLiPSe aunque IF/Prolog no tuvo buenos resultados en una de las pruebas. Los dos sistemas restantes, (ECLiPSe y CHR), resultaron ser los más lentos, aunque CHR tiene como ventaja que está diseñado para definir solucionadores de restricciones adecuados para problemas particulares en dominios específicos. Como conclusión final, los autores resumen que para obtener la máxima eficiencia ILOG Solver es la mejor alternativa, mientras que clp (FD) también es una buena opción mientras que el problema a resolver sea bastante pequeño, (en términos de variables). Se destaca que CHR, como se mencionó anteriormente, es la mejor alternativa desde un punto de vista expresivo.

Se repasa a continuación los solucionadores más conocidos.

Gecode, [38], es un conjunto de herramientas C++ de código abierto para desarrollar sistemas y aplicaciones basados en restricciones. Gecode proporciona un solucionador de restricciones con un rendimiento de vanguardia a la vez que es modular y extensible. Entre sus ventajas podemos encontrar que se puede conectar fácilmente a otros sistemas, admite la programación de nuevas restricciones, estrategias de ramificación y motores de búsqueda, es completo; posee características tales como diversos motores de búsqueda, ruptura automática de simetría, (LDSB), heurística de ramificación avanzada, entre otros. Además, ofrece un rendimiento excelente con respecto al tiempo de ejecución y al uso de la memoria; ganó todas las medallas de oro en todas las categorías de los Desafíos MiniZinc desde 2008 a 2012. El software incluye un tutorial completo, es gratis y está probado con más de 50.000 casos de prueba, alcanzando una cobertura de prueba cercana al 100%.

Por otro lado, Choco solver es otro software disponible: es una biblioteca que utiliza lenguaje Java, hecha para resolver problemas de Programación con Restricciones, [39]. El usuario modela su problema, estableciendo el conjunto de restricciones que deben satisfacerse en cada solución. Luego, el problema se resuelve alternando algoritmos de filtrado de restricciones con un mecanismo de búsqueda.

La primera versión de Choco data de principios de la década de 2000. Unos años más tarde, Choco 2 tuvo un gran éxito tanto en el mundo académico como en el industrial. Por cuestiones de mantenimiento, Choco se reescribió por completo en 2011, lo que llevó a crear Choco 3. La primera versión beta de Choco 3 se lanzó en 2012 y mostró una mejora significativa en el rendimiento. Luego, Choco 4 vino con una API de modelado más simple. La última versión es Choco 4.10.5, cuyo lanzamiento fue el 2 de octubre del presente año.

Choco se encuentra entre los solucionadores de Programación con Restricciones más rápidos del mercado. En 2013 y 2014, Choco recibió dos medallas de plata y tres medallas de bronce en el desafío MiniZinc, que es la competencia mundial de programadores de restricciones. Además de estos resultados de desempeño, Choco se beneficia de colaboradores académicos, que brindan apoyo y mejoras a largo plazo, y de la empresa consultora COSLING, que brinda servicios que van desde la capacitación hasta el desarrollo y la integración de modelos de PC en aplicaciones más grandes.
Asimismo, [40], Oz se basa en la programación funcional de orden superior y la programación lógica con restricciones; combina funciones con relaciones. Oz proporciona algoritmos para decidir la satisfacibilidad e implementación de restricciones básicas con la siguiente forma: X = n; X = Y o X: D donde X e Y son variables, n es un entero no negativo y D es un dominio finito. Las restricciones básicas residen en el almacén de restricciones. Las restricciones no básicas, como X + Y = Z, no están contenidas en el almacén, pero son impuestas por propagadores, siendo estos agentes que leen en el almacén de restricciones e intentan reducir los dominios fijados ahí añadiendo restricciones básicas al almacén. Oz tiene una semántica formal simple y una implementación eficiente, el Sistema de Programación Mozart. Oz es un lenguaje orientado a la concurrencia, (un sistema concurrente es aquel en el que un cálculo puede avanzar sin esperar a que se completen todos los demás cálculos). Pertenece a la familia de los CCPLs, (Concurrent Constraint Programming Languages, CCPLs).

IBM ILOG CPLEX Optimization Studio, [41], utiliza tecnología de optimización de decisiones para optimizar las decisiones de negocio, desarrollar y desplegar modelos de optimización en forma rápida, y crear aplicaciones del mundo real que puedan mejorar significativamente el resultado del negocio. Es una solución de análisis prescriptivo que permite un rápido desarrollo y despliegue de modelos de optimización de decisiones utilizando programación matemática y de restricciones. Combina un entorno de desarrollo integrado con todas las funciones que admite el lenguaje de programación de optimización, (OPL) y los solucionadores CPLEX y CP Optimizer de alto rendimiento.

A su vez, IBM ILOG CP Optimizer, es un complemento para los especialistas en optimización para resolver problemas de planificación y programación operativa del mundo real. Contiene un optimizador robusto que maneja las limitaciones que invariablemente se encuentran en tales desafíos. Para problemas puramente académicos, (por ejemplo, talleres de trabajo, talleres abiertos y talleres de flujo), encuentra soluciones comparables a las encontradas por algoritmos especializados de última generación.

Ciertos problemas de optimización combinatoria no se pueden linealizar y resolver fácilmente con los métodos tradicionales de Programación Matemática. Para manejar estos problemas, ILOG CP Optimizer proporciona un gran conjunto de restricciones aritméticas y lógicas, así como un optimizador robusto que trae todos los beneficios de un proceso de desarrollo de modelo y ejecución a la optimización combinatoria.

Según [42], IBM ILOG CP Optimizer es un sistema genérico basado en programación con restricciones para modelar y resolver problemas de programación. Proporciona un lenguaje

algebraico con conceptos matemáticos simples para capturar la dimensión temporal de los problemas de programación en un marco de trabajo de optimización combinatoria. CP Optimizer implementa un paradigma de modelado y ejecución que reduce enormemente la carga para el usuario al momento de comprender la programación con restricciones en sí misma o los algoritmos de programación: el modelado es, sin duda, el más importante. La búsqueda automática proporciona un buen rendimiento desde el primer momento y está mejorando continuamente.

Finalmente, [43] ECLiPSe es un entorno de programación que contiene el lenguaje de programación PROLOG y algunas extensiones que permitirán manejar bases de datos, (incluyendo bases de datos declarativas y bases de conocimiento), programación lógica basada en restricciones (en concreto programación lógica basada en restricciones sobre dominios finitos), y programación concurrente. Eclipse incluye las restricciones en dominio finito tradicionales. Permite además la escritura de extensiones como restricciones definidas por el usuario o solvers completamente nuevos como CHR. Eclipse fue creado en 1991, y estaba inicialmente basado en el lenguaje de programación CHIP. En los siguientes 15 años, los solvers de Programación con Restricciones y las interfaces de los solvers soportados por Eclipse fueron continuamente extendidos para responder a los requerimientos de los usuarios. La integración de solver sobre dominios finitos y solvers de Programación Lineal, soportando algoritmos híbridos se realizó en el año 2000. En 2001 se creó la librería IC, la cual soporta restricciones booleanas, enteras y reales. También incluye las librerías de algunos otros lenguajes de programación lógica, por ejemplo, CLP(R), que fue desarrollado de forma separada. En agosto de 2004, Eclipse fue adquirida por Cisco System. Es utilizado por múltiples instituciones para educación e investigación por todo el mundo, siendo explotada para múltiples propósitos: planeación de producción, scheduling, biomática, entre otros.

4. Conclusiones

Este documento presenta el relevamiento y análisis de la literatura orientada a Problemas de Planificación aplicado en diversas actividades que tiene que ver con Asignación de tareas, Asignación de recursos, Planificación de calendarios entre otros.

Se orientó la búsqueda en Programación con Restricciones para, de esta forma, poder determinar en qué etapa de desarrollo se encuentra esta área. De igual forma, se realizó una investigación de las herramientas utilizadas en los últimos años para resolver problemas de esta índole mediante esta técnica.

La Programación con Restricciones es un paradigma de la programación, en el cual las relaciones entre las variables se expresan en términos de restricciones. Se utiliza normalmente para modelar problemas combinatorios especialmente difíciles. Se pudo comprobar en la literatura relevada, que el uso de la Programación con Restricciones aplicada a los problemas de planificación demuestra un rendimiento eficiente para abordar problemas de este tipo. Además, es flexible y permite absorber modificaciones en las condiciones o variables del problema a modelar, en caso de ser necesario. Si bien la búsqueda se orientó a la aplicación de esta herramienta en la temática previamente mencionada, se encontró que es también sumamente utilizada en otras áreas tales como inteligencia artificial, (siendo éste el sector pionero donde se utilizó la herramienta), bases de datos e investigación operativa.

Se puede afirmar que el uso de la herramienta potencia considerablemente la gestión en las diferentes áreas en las cuales es aplicada, trayendo como consecuencia un mejor control de las variables intrínsecas de los procesos de planificación, evidenciando controles más rápidos, flexibilidad, optimización, mayor rango de acción a la hora de tomar decisiones y la oportunidad de simular diferentes escenarios en función de los requerimientos.

Asimismo, esta herramienta nos permite relacionar variables que en principio son de carácter independiente pero que a través de la misma es posible considerarlas de manera simultánea, obteniendo así una solución que optimiza a este grupo de variables, siendo éste el objetivo principal del modelado.

Por otro lado, es de consideración destacar que, si bien una de las ventajas de la Programación con Restricciones es que es posible relacionar gran cantidad de variables, existen ciertas limitaciones respecto al tamaño de los diferentes modelos: a mayor número de variables y/o restricciones, mayor tiempo requerido para la solución computacional y a su vez los requerimientos de almacenamiento de datos dentro del modelo. A pesar de hoy día existir software de vanguardia, éstos tienen limitaciones con respecto al tamaño de los modelos.

Se ha constatado en varios trabajos el abordaje de problemas utilizando modelos híbridos en donde se combina la Programación con Restricciones con Programación lineal, Técnicas de Investigación de Operaciones, Enteros mixtos y Metaheurísticas, logrando buenos resultados en términos de una solución factible en problemas donde, de otro modo no resultaría eficiente la solución encontrada, o incluso no se logra obtener una solución.

Bibliografía

[1] Networking.
https://networkingrd.net/2019/10/16/concepto-planificacion-que-es-y-para-que-sirve/
Último acceso: 23/05/2020.

[2] Diccionario Real Academia Española. https://dle.rae.es/evento
Último acceso: 23/05/2020.

[3] Programación con Restricciones, Wikipedia.
https://en.wikipedia.org/wiki/Constraint_programming
Último acceso: 15/06/2020.

[4] J. A. Gutiérrez Quezada, A. E. López Estay. Tesis de grado, Escuela de Ingeniería Informática, Facultad de Ingeniería, Pontificia Universidad Católica de Valparaíso, 2011.

[5] Plataforma Timbó. http://www.timbo.org.uy.
Último acceso: 21/07/2020.

[6] J. Hamiez, J. Hao. A note on a sports league scheduling problem, 2014.

[7] A. Aggoun, A. Vazacopoulos. Solving Sports Scheduling and Timetabling Problems with Constraint Programming. *Economics, Management and Optimization in Sports*, 243-264, 2004.

[8] R. Russell, T. Urban. A constraint programming approach to the multiple-venue, sport-scheduling problem. *Computers & Operations Research 33*, 1895-1906, 2006.

[9] B. M. Smith, S. C. Brailsford, P. M. Hubbard, H. P. Williams. The progressive party problem: Integer linear programming and constraint programming compared. Principles and Practice of Constraint Programming. *Lecture Notes in Computer Science 976*, 36-52, 1995.

[10] T. Benoist, E. Bourreau, B. Rottembourg. The TV-Break Packing Problem. *European Journal of Operational Research Volume 176, Issue 3,* 1 February 2007, Pages 1371-1386, 2007.

[11] S. Deris, S. Omatu, H. Ohta. Timetable planning using the constraint-based reasoning, *Computers & Operations Research* 27, 819-840, 2000.

[12] C. Valouxis, E. Housos. Constraint programming approach for school timetabling. *Computers & Operations Research 30*, 1555-1572, 2003.

[13] M. Sellmann, K. Zervoudakis, P. Stamatopoulos. Crew Assignment via Constraint Programming: Integrating Column Generation and Heuristic Tree Search. *Annals of Operations Research 115*, 207–225, 2002.

[14] M. A. Ornek, C. Ozturk, I. Sugut. Integer and constraint programming model formulations for flight-gate assignment problem. *Oper Res Int J*, 2020.

[15] S. Gabteni, M. Grönkvist. Combining column generation and constraint programming to solve the tail assignment problem. *Ann Oper Res 171*, 61, 2009.

[16] F. Martin, A. Pinkney, X. Yu. Cane Railway Scheduling via Constraint Logic Programming: Labelling Order and Constraints in a Real-Life Application. *Annals of Operations Research 108*, 193–209, 2001.

[17] A. J. Pinkney. An automatic cane railway scheduling system, Masters thesis, James Cook University of North Queensland, 1988.

[18] N. El Hachemi, M. Gendreau, L.M. Rousseau. Solving a Log-Truck Scheduling Problem with Constraint Programming. *Lecture Notes in Computer Science 5015*, 293-297, 2008.

[19] A. de Silva. Combining Constraint Programming and Linear Programming on an Example of Bus Driver Scheduling Programming on an Example of Bus Driver Scheduling. *Annals of Operations Research 108*, 277–291, 2001.

[20] A. F. Han, E. C. Li. A constraint programming-based approach to the crew scheduling problem of the Taipei mass rapid transit system. *Ann Oper Res 223*, 173– 193, 2014.

[21] L. D. Gaspero, A. Rendl, T. Urli. Balancing bike sharing systems with constraint programming. *Constraints 21*, 318–348 (2016).

[22] L. D. Gaspero, A. Rendl, T. Urli. Constraint-based approaches for balancing bike sharing systems. *Lecture Notes in Computer Science 8124,* 758–773, 2013.

[23] L. D. Gaspero, A. Rendl, T. Urli. A hybrid ACO + CP for balancing bicycle sharing systems. *Lecture Notes in Computer Science 7919*, 198–212, 2013.

[24] Y. Chen, Z. Guan, Y. Peng. Technology and system of constraint programming for industry production scheduling — Part I: A brief survey and potential directions. *Frontiers in Mechanical Engineering in China 5*, 455–464, 2010.

[25] K. Limtanyakul. Scheduling of Tests on Vehicle Prototypes Using Constraint and Integer Programming. *Operations Research Proceedings 2007*, 421-426, 2008.

[26] M. Sirolla, J. M. Novas, G. P. Henning. Programación de la producción a corto plazo y de tareas de mantenimiento preventivo en ambientes job shop flexibles. *Iberoamerican Journal of Industrial Engineering 8*, 192-207, 2016.

[27] C. Timpe. Solving planning and scheduling problems with combined integer and constraint programming. *OR Spectrum 24*, 431–448, 2002.

[28] J. N. Hooker. A Hybrid Method for the Planning and Scheduling. *Constraints 10*, 385–401, 2005.

[29] E. B. Edis. Constraint programming approaches to disassembly line balancing problem with sequencing decisions. *Computers and Operations Research 126*, 2021.

[30] F. M. Novara, G. P. Henning. Planeación reactiva de plantas “batch” multiproducto, multietapa basada en programación con restricciones. *Iberoamerican Journal of Industrial Engineering 6(11)*, 294-310, 2014.

[31] M. Wallace. Practical applications of constraint programming. *Constraints 1*, 139-168, 1996.

[32] M. Milano, M. Wallacez. Integrating operations research in constraint programming. *4 OR* 4, 175–219, 2006.

[33] A. Elkhyari, C. Guéret, N. Jussien. Solving Dynamic Resource Constraint Project Scheduling Problems Using New Constraint Programming Tools. *Practice and Theory of Automated Timetabling IV*, 39-59, 2002.

[34] K. Darby-Dowman, J. Little, G. Mitra. Constraint Logic Programming and Integer Programming approaches and their collaboration in solving an assignment scheduling problem. *Constraints 1*, 245–264, 1997.

[35] W. Tao, N. Meskens, D. Duvivier. Scheduling operating theatres: Mixed integer programming vs. constraint programming. *European Journal of Operational Research 247*, 401–413, 2015.

[36] N. Meskens, D. Duvivier, A. Hanset. Multiobjective scheduling of operating theatre considering desiderata of surgical team. *Decision Support Systems 55*, 650-659, 2013.

[37] A. J. Fernández, P. M. Hill. A Comparative Study of Eight Constraint Programming Languages Over the Boolean and Finite Domains. *Constraints 5*, 275–301, 2000.

[38] C. Schulte, G. Tack, M. Z. Lagerkvist. *Modeling and Programming with Gecode,* 2019.

[39] C. Prud'homme, J. G. Fages, X. Lorca. *Choco Solver Documentation Release 4.0.8,* 2018.

[40] J. Parejas, F. Andrés. Tesis de grado, Escuela de Ingeniería Informática, Facultad de Ingeniería, Pontificia Universidad Católica de Valparaíso, 2008.

[41] Sitio web oficial de IBM. https://www.ibm.com/analytics/cplex-cp-optimizer Último acceso: 19/07/2020.

[42] P. Laborie, J. Rogerie, P. Shaw. IBM ILOG CP optimizer for scheduling. *Constraints 23*, 210–250, 2018.

[43] M. Teresa Escrig. *El Lenguaje de Programación PROLOG*, 2001.

[44] F. Barber, M. Salido. Introducción a la Programación de Restricciones, *Revista Iberoamericana de Inteligencia Artificial 7*, 13-30, 2003.

[45] S. Lizama, A. Muñoz. Algoritmos de Consistencia para Programación con Restricciones, 2013.

ANEXO 2 - Resultados de la encuesta de convocatoria realizada en 2019

	CONJUNTOS	CATEGORIA	CONVOCATORIA ALTA	CONVOCATORIA MEDIA	CONVOCATORIA BAJA	PONDERACIÓN DE CONVOCATORIA
1	SAPHIRUS	REVISTA	24	6	0	9,00
2	ADRENALINA	REVISTA	24	7	0	8,87
3	DULCINEA	REVISTA	22	4	1	8,93
4	ZODIACO	REVISTA	9	18	2	6,28
5	L.BUSCADORES	REVISTA	4	15	11	4,2
6	FENIX	REVISTA	21	2	4	8,3
7	JADE	REVISTA	0	9	21	2,2
8	MANDALA	REVISTA	2	4	16	2,55
9	QUIJOTES	PARODISTAS	26	1	1	9,5
10	BUBY'S BIS	PARODISTAS	22	5	1	8,79
11	POPPIN'S	PARODISTAS	19	9	0	8,39
12	POPPIN'S	PARODISTAS	13	14	0	7,41
13	POPPIN'S	PARODISTAS	1	16	9	3,81
14	POPPIN'S	PARODISTAS	1	7	16	2,54
15	POPPIN'S	PARODISTAS	1	19	7	4,15
16	POPPIN'S	PARODISTAS	4	11	5	5
17	POPPIN'S	PARODISTAS	3	7	10	3,75
18	POPPIN'S	PARODISTAS	2	7	11	3,3
19	POPPIN'S	PARODISTAS	2	7	13	3,09
20	POPPIN'S	PARODISTAS	1	5	13	2,53
21	POPPIN'S	HUMORISTAS	18	7	1	8,31
22	POPPIN'S	HUMORISTAS	11	14	1	6,96
23	POPPIN'S	HUMORISTAS	16	6	3	7,72
24	POPPIN'S	HUMORISTAS	20	6	0	8,85
25	POPPIN'S	HUMORISTAS	2	9	12	3,35
26	POPPIN'S	HUMORISTAS	2	12	10	3,75
27	POPPIN'S	HUMORISTAS	4	9	13	3,77
28	POPPIN'S	HUMORISTAS	7	10	8	5,12
29	POPPIN'S	HUMORISTAS	1	4	17	2,14
30	POPPIN'S	HUMORISTAS	1	3	12	2,31
31	POPPIN'S	HUMORISTAS	1	0	28	1,31
32	POPPIN'S	MURGA	28	3	0	9,52
33	POPPIN'S	MURGA	27	4	0	9,35
34	POPPIN'S	MURGA	3	21	4	4,96
35	POPPIN'S	MURGA	22	5	1	8,79
36	POPPIN'S	MURGA	7	17	5	5,52
37	POPPIN'S	MURGA	8	13	8	5,28
38	POPPIN'S	LUBOLOS	12	8	12	5,38

39	POPPIN'S	LUBOLOS	2	4	18	2,42
40	POPPIN'S	LUBOLOS	4	5	16	3,24

ANEXO 3 - Ponderaciones por agrupación por escenario-instancia

NRO.	AGRUPACIÓN	CATEGORÍA	HORARIO	POND. 1.1	POND. 2.1	POND. 2.2	POND. 3.1	POND. 3.2
1	SAPHIRUS	REVISTA	CUARTA	9,00	9,00	10,00	10	1
2	ADRENALINA	REVISTA	CUARTA	8,87	8,87	10,00	10	1
3	DULCINEA	REVISTA	CUARTA	8,93	8,93	10,00	10	1
4	ZODIACO	REVISTA	SEGUNDA	6,28	6,28	6,28	10	1
5	L. BUSCADORES	REVISTA	CUARTA	4,20	4,20	10,00	10	1
6	FENIX	REVISTA	SEGUNDA	8,30	8,30	8,30	10	1
7	JADE	REVISTA	SEGUNDA	2,20	2,20	2,20	10	1
8	MANDALA	REVISTA	SEGUNDA	2,55	2,55	2,55	10	1
9	QUIJOTES	PARODISTAS	CUARTA	9,50	9,50	10,00	10	1
10	BUBY'S BIS	PARODISTAS	CUARTA	8,79	8,79	10,00	10	1
11	ZERO	PARODISTAS	PRIMERA	6,20	10,00	6,20	10	1
12	CHERRY'S	PARODISTAS	PRIMERA	4,00	10,00	4,00	10	1
13	TROYANOS	PARODISTAS	TERCERA	4,15	4,15	4,15	10	1
14	WONKA'S	PARODISTAS	CUARTA	5,00	5,00	10,00	10	1
15	CELESTINOS	PARODISTAS	TERCERA	3,75	3,75	3,75	10	1
16	PRINCIPES	PARODISTAS	TERCERA	3,30	3,30	3,30	10	1
17	TOON'S	PARODISTAS	TERCERA	3,09	3,09	3,09	10	1
18	ZABRITOS	PARODISTAS	TERCERA	2,53	2,53	2,53	10	1
19	FRENESI	HUMORISTAS	PRIMERA	4,30	10,00	4,30	10	1
20	VALU'S	HUMORISTAS	TERCERA	6,96	6,96	6,96	10	1
21	GNOMOS	HUMORISTAS	TERCERA	7,72	7,72	7,72	10	1
22	COLIBRIQUIS	HUMORISTAS	PRIMERA	5,80	10,00	5,80	10	1
23	CACHIRULOS	HUMORISTAS	TERCERA	3,35	3,35	3,35	10	1
24	HEREDEROS	HUMORISTAS	PRIMERA	3,70	10,00	3,70	10	1
25	DEPORTADOS	HUMORISTAS	PRIMERA	2,60	10,00	2,60	10	1
26	BAM BAM	HUMORISTAS	TERCERA	5,12	5,12	5,12	10	1
27	YOGUIS	HUMORISTAS	PRIMERA	4,60	10,00	4,60	10	1
28	LOS CHAPITAS	HUMORISTAS	TERCERA	2,31	2,31	2,31	10	1
29	LOS TOBY'S	HUMORISTAS	PRIMERA	1,31	10,00	1,31	10	1
30	DESCARA2	HUMORISTAS	PRIMERA	2,10	10,00	2,10	10	1
31	MANO A MANO	MURGA	CUARTA	9,52	9,52	10,00	10	1
32	LA ZAFADA	MURGA	CUARTA	9,35	9,35	10,00	10	1
33	LOS DUENDES	MURGA	SEGUNDA	7,20	7,20	7,20	10	1
34	L. PEPINITOS	MURGA	CUARTA	8,79	8,79	10,00	10	1
35	LA DESCOCADA	MURGA	SEGUNDA	5,52	5,52	5,52	10	1
36	DIABLITOS VERDES	MURGA	SEGUNDA	5,28	5,28	5,28	10	1
37	DON BARULLO	MURGA	SEGUNDA	6,50	6,50	6,50	10	1
38	COSME Y DAMIAN	LUBOLOS	SEGUNDA	7,40	7,40	7,40	10	1
39	S.C.	LUBOLOS	PRIMERA	3,20	10,00	3,20	10	1
40	OHANA	LUBOLOS	SEGUNDA	3,24	3,24	3,24	10	1
			w	**5,41**	**6,97**	**5,86**	**10**	**1**

NRO.	AGRUPACIÓN	CATEGORÍA	HORARIO	POND. 4.1	POND. 4.2	POND. 4.3	POND. 4.4
1	SAPHIRUS	REVISTA	CUARTA	5,84	7,47	1,49	7,08
2	ADRENALINA	REVISTA	CUARTA	1,97	9,91	6,45	6,14
3	DULCINEA	REVISTA	CUARTA	1,83	7,94	9,32	8,42
4	ZODIACO	REVISTA	SEGUNDA	6,10	2,17	2,04	6,65
5	L. BUSCADORES	REVISTA	CUARTA	8,68	7,84	6,87	5,26
6	FENIX	REVISTA	SEGUNDA	6,80	1,29	8,95	6,65
7	JADE	REVISTA	SEGUNDA	5,95	3,75	2,44	5,72
8	MANDALA	REVISTA	SEGUNDA	1,48	7,82	1,49	9,58
9	QUIJOTES	PARODISTAS	CUARTA	1,28	8,10	1,60	4,17
10	BUBY'S BIS	PARODISTAS	CUARTA	3,49	1,88	8,61	4,02
11	ZERO	PARODISTAS	PRIMERA	2,56	2,85	2,73	2,73
12	CHERRY'S	PARODISTAS	PRIMERA	2,79	7,52	1,11	7,77
13	TROYANOS	PARODISTAS	TERCERA	4,82	1,35	1,70	2,25
14	WONKA'S	PARODISTAS	CUARTA	9,31	9,58	5,95	8,05
15	CELESTINOS	PARODISTAS	TERCERA	9,86	1,68	9,27	7,61
16	PRINCIPES	PARODISTAS	TERCERA	4,62	6,88	4,48	4,54
17	TOON'S	PARODISTAS	TERCERA	4,32	5,29	3,03	7,42
18	ZABRITOS	PARODISTAS	TERCERA	3,00	5,90	8,61	3,97
19	FRENESI	HUMORISTAS	PRIMERA	7,38	5,18	9,54	2,53
20	VALU'S	HUMORISTAS	TERCERA	7,79	5,57	6,14	4,24
21	GNOMOS	HUMORISTAS	TERCERA	9,74	5,98	6,33	3,30
22	COLIBRIQUIS	HUMORISTAS	PRIMERA	6,93	7,19	9,57	2,01
23	CACHIRULOS	HUMORISTAS	TERCERA	4,79	1,28	5,31	2,85
24	HEREDEROS	HUMORISTAS	PRIMERA	6,59	2,36	3,49	6,15
25	DEPORTADOS	HUMORISTAS	PRIMERA	8,28	3,92	9,50	8,55
26	BAM BAM	HUMORISTAS	TERCERA	8,56	4,61	9,47	2,46
27	YOGUIS	HUMORISTAS	PRIMERA	3,72	3,16	2,64	5,87
28	LOS CHAPITAS	HUMORISTAS	TERCERA	2,34	8,36	5,36	6,59
29	LOS TOBY'S	HUMORISTAS	PRIMERA	8,15	2,20	1,61	9,62
30	DESCARA2	HUMORISTAS	PRIMERA	7,46	5,75	6,67	3,50
31	MANO A MANO	MURGA	CUARTA	4,68	6,12	7,07	1,19
32	LA ZAFADA	MURGA	CUARTA	7,20	6,95	3,33	4,77
33	LOS DUENDES	MURGA	SEGUNDA	3,74	5,75	1,68	5,04
34	L. PEPINITOS	MURGA	CUARTA	3,82	3,78	6,76	7,04
35	LA DESCOCADA	MURGA	SEGUNDA	7,74	2,22	2,61	4,06
36	DIABLITOS VERDES	MURGA	SEGUNDA	3,40	2,69	6,06	1,90
37	DON BARULLO	MURGA	SEGUNDA	8,98	7,66	8,46	8,23
38	COSME Y DAMIAN	LUBOLOS	SEGUNDA	7,24	1,52	1,98	4,53
39	S.C.	LUBOLOS	PRIMERA	4,10	3,55	8,48	8,34
40	OHANA	LUBOLOS	SEGUNDA	2,21	8,95	1,88	8,41
			w	**5,49**	**5,1**	**5,25**	**5,48**

ANEXO 4 - Fixtures obtenidos en la experimentación numérica, casos 1.1 y 4.1.

Modelo R1

1.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	S.C.	LOS TOBY'S	DESCARA2	CHERRY'S	ZERO	HEREDEROS	FRENESI	DEPORTADOS	YOGUIS	COLIBRIQUIS
2	JADE	COSME Y DAMIAN	FENIX	MANDALA	LOS DUENDES	DON BARULLO	LA DESCOCADA	ZODIACO	DIABLITOS VERDES	OHANA
3	GNOMOS	TROYANOS	LOS CHAPITAS	BAM BAM	CACHIRULOS	VALU'S	ZABRITOS	CELESTINOS	TOON'S	PRINCIPES
4	BUBY'S BIS	ADRENALINA	QUIJOTES	LA ZAFADA	LOS BUSCADORES	WONKA'S	DULCINEA	MANO A MANO	SAPHIRUS	LOS PEPINITOS
Prom. por etapa	**5,48**	**5,43**	**5,55**	**5,26**	**5,24**	**5,54**	**5,32**	**5,54**	**5,49**	**5,28**

HUMORISTAS
REVISTAS
PARODISTAS
MURGA
LUBOLOS

1.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	S.C.	LOS TOBY'S	YOGUIS	ZERO	COLIBRIQUIS	CHERRY'S	HEREDEROS	DESCARA2	DEPORTADOS	FRENESI
2	DON BARULLO	FENIX	LA DESCOCADA	MANDALA	COSME Y DAMIAN	OHANA	DIABLITOS VERDES	JADE	LOS DUENDES	ZODIACO
3	VALU'S	PRINCIPES	ZABRITOS	CACHIRULOS	TROYANOS	BAM BAM	CELESTINOS	GNOMOS	TOON'S	LOS CHAPITAS
4	WONKA'S	LOS PEPINITOS	SAPHIRUS	MANO A MANO	LOS BUSCADORES	LA ZAFADA	DULCINEA	QUIJOTES	ADRENALINA	BUBY'S BIS
Prom. por etapa	**5,42**	**5,43**	**5,41**	**5,41**	**5,39**	**5,43**	**5,42**	**5,38**	**5,44**	**5,42**

HUMORISTAS
REVISTAS
PARODISTAS
MURGA
LUBOLOS

4.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	S.C.	ZERO	DEPORTADOS	DESCARA2	LOS TOBY'S	HEREDEROS	YOGUIS	COLIBRIQUIS	CHERRY'S	FRENESI
2	ZODIACO	DIABLITOS VERDES	MANDALA	OHANA	JADE	FENIX	LA DESCOCADA	COSME Y DAMIAN	DON BARULLO	LOS DUENDES
3	LOS CHAPITAS	VALU'S	TROYANOS	CELESTINOS	TOON'S	CACHIRULOS	PRINCIPES	ZABRITOS	BAM BAM	GNOMOS
4	WONKA'S	LOS BUSCADORES	LA ZAFADA	ADRENALINA	LOS PEPINITOS	BUBY'S BIS	SAPHIRUS	MANO A MANO	DULCINEA	QUIJOTES
Prom. por etapa	**4,20**	**5,66**	**4,66**	**4,49**	**3,85**	**6,04**	**5,61**	**6,31**	**6,14**	**7,18**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	LOS TOBY'S	FRENESI	HEREDEROS	CHERRY'S	S.C.	YOGUIS	DEPORTADOS	DESCARA2	COLIBRIQUIS	ZERO
2	OHANA	MANDALA	DON BARULLO	COSME Y DAMIAN	ZODIACO	LA DESCOCADA	LOS DUENDES	JADE	DIABLITOS VERDES	FENIX
3	ZABRITOS	GNOMOS	TOON'S	CACHIRULOS	LOS CHAPITAS	PRINCIPES	BAM BAM	TROYANOS	CELESTINOS	VALU'S
4	LOS BUSCADORES	BUBY'S BIS	ADRENALINA	LA ZAFADA	WONKA'S	SAPHIRUS	QUIJOTES	LOS PEPINITOS	DULCINEA	MANO A MANO
Prom. por etapa	**2,82**	**5,84**	**5,54**	**6,03**	**4,20**	**5,61**	**6,11**	**4,31**	**5,94**	**7,75**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Modelo R2

1.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	HEREDEROS	ZERO	DESCARA2	YOGUIS	S.C.	LOS TOBY'S	COLIBRIQUIS	DEPORTADOS	CHERRY'S	FRENESI
2	DON BARULLO	MANDALA	LA DESCOCADA	LOS DUENDES	ZODIACO	FENIX	DIABLITOS VERDES	OHANA	COSME Y DAMIAN	JADE
3	TOON'S	LOS CHAPITAS	CACHIRULOS	ZABRITOS	BAM BAM	CELESTINOS	PRINCIPES	GNOMOS	VALU'S	TROYANOS
4	SAPHIRUS	LOS PEPINITOS	BUBY'S BIS	ADRENALINA	WONKA'S	LA ZAFADA	DULCINEA	QUIJOTES	LOS BUSCADORES	LOS BUSCADORES
Prom. por etapa	**5,57**	**4,96**	**4,94**	**5,80**	**4,90**	**5,68**	**5,83**	**5,77**	**5,64**	**3,71**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	CACHIRULOS	LOS CHAPITAS	BAM BAM	TOON'S	ZABRITOS	CELESTINOS	VALU'S	TROYANOS	PRINCIPES	GNOMOS
2	DON BARULLO	LA DESCOCADA	MANDALA	ZODIACO	LOS DUENDES	FENIX	JADE	OHANA	COSME Y DAMIAN	DIABLITOS VERDES
3	YOGUIS	ZERO	S.C.	HEREDEROS	DESCARA2	LOS TOBY'S	FRENESI	COLIBRIQUIS	DEPORTADOS	CHERRY'S
4	BUBY'S BIS	SAPHIRUS	WONKA'S	LOS PEPINITOS	ADRENALINA	LA ZAFADA	QUIJOTES	DULCINEA	MANO A MANO	LOS BUSCADORES
Prom. por etapa	**5,81**	**5,76**	**3,97**	**5,47**	**5,18**	**5,68**	**5,74**	**5,53**	**5,71**	**5,30**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

1.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	FRENESI	CHERRY'S	ZERO	YOGUIS	S.C.	DEPORTADOS	COLIBRIQUIS	HEREDEROS	LOS TOBY'S	DESCARA2
2	ZODIACO	DON BARULLO	MANDALA	LA DESCOCADA	FENIX	LOS DUENDES	OHANA	DIABLITOS VERDES	COSME Y DAMIAN	JADE
3	LOS CHAPITAS	VALU'S	CACHIRULOS	ZABRITOS	BAM BAM	TOON'S	PRINCIPES	CELESTINOS	TROYANOS	GNOMOS
4	BUBY'S BIS	LOS BUSCADORES	MANO A MANO	DULCINEA	WONKA'S	ADRENALINA	LA ZAFADA	SAPHIRUS	LOS PEPINITOS	QUIJOTES
Prom. por etapa	**5,42**	**5,42**	**5,41**	**5,40**	**5,41**	**5,44**	**5,42**	**5,43**	**5,41**	**5,38**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	ZABRITOS	CACHIRULOS	LOS CHAPITAS	VALU'S	PRINCIPES	TOON'S	BAM BAM	TROYANOS	GNOMOS	CELESTINOS
2	LA DESCOCADA	MANDALA	ZODIACO	DON BARULLO	OHANA	LOS DUENDES	FENIX	COSME Y DAMIAN	JADE	DIABLITOS VERDES
3	YOGUIS	ZERO	FRENESI	CHERRY'S	COLIBRIQUIS	DEPORTADOS	S.C.	LOS TOBY'S	DESCARA2	HEREDEROS
4	DULCINEA	MANO A MANO	BUBY'S BIS	LOS BUSCADORES	LA ZAFADA	ADRENALINA	WONKA'S	LOS PEPINITOS	QUIJOTES	SAPHIRUS
Prom. por etapa	**5,40**	**5,41**	**5,42**	**5,42**	**5,42**	**5,44**	**5,41**	**5,41**	**5,38**	**5,43**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	HEREDEROS	FRENESI	LOS TOBY'S	YOGUIS	CHERRY'S	ZERO	DESCARA2	S.C.	COLIBRIQUIS	DEPORTADOS
2	MANDALA	LOS DUENDES	JADE	ZODIACO	DON BARULLO	OHANA	DIABLITOS VERDES	FENIX	LA DESCOCADA	COSME Y DAMIAN
3	CELESTINOS	PRINCIPES	TOON'S	LOS CHAPITAS	BAM BAM	GNOMOS	ZABRITOS	VALU'S	TROYANOS	CACHIRULOS
4	MANO A MANO	SAPHIRUS	LOS PEPINITOS	WONKA'S	DULCINEA	LA ZAFADA	LOS BUSCADORES	BUBY'S BIS	ADRENALINA	QUIJOTES
Prom. por etapa	**4,88**	**5,95**	**3,85**	**4,55**	**6,14**	**6,63**	**3,53**	**6,81**	**6,09**	**5,71**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	CELESTINOS	TOON'S	BAM BAM	PRINCIPES	GNOMOS	LOS CHAPITAS	VALU'S	TROYANOS	CACHIRULOS	ZABRITOS
2	JADE	LOS DUENDES	MANDALA	ZODIACO	OHANA	FENIX	DON BARULLO	DIABLITOS VERDES	COSME Y DAMIAN	LA DESCOCADA
3	YOGUIS	LOS TOBY'S	FRENESI	HEREDEROS	CHERRY'S	ZERO	S.C.	COLIBRIQUIS	DEPORTADOS	DESCARA2
4	MANO A MANO	SAPHIRUS	WONKA'S	LOS PEPINITOS	DULCINEA	LA ZAFADA	QUIJOTES	LOS BUSCADORES	BUBY'S BIS	ADRENALINA
Prom. por etapa	**5,02**	**5,15**	**4,24**	**5,52**	**5,97**	**6,54**	**6,54**	**4,86**	**5,54**	**4,76**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	CHERRY'S	COLIBRIQUIS	DESCARA2	HEREDEROS	FRENESI	LOS TOBY'S	S.C.	YOGUIS	ZERO	DEPORTADOS
2	COSME Y DAMIAN	DIABLITOS VERDES	JADE	DON BARULLO	MANDALA	OHANA	ZODIACO	LA DESCOCADA	FENIX	LOS DUENDES
3	CACHIRULOS	ZABRITOS	TROYANOS	TOON'S	GNOMOS	CELESTINOS	LOS CHAPITAS	PRINCIPES	VALU'S	BAM BAM
4	LA ZAFADA	LOS BUSCADORES	LOS PEPINITOS	ADRENALINA	BUBY'S BIS	DULCINEA	WONKA'S	SAPHIRUS	MANO A MANO	QUIJOTES
Prom. por etapa	**6,03**	**4,45**	**4,31**	**5,54**	**5,84**	**4,31**	**4,20**	**5,61**	**7,75**	**6,11**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	TROYANOS	ZABRITOS	CACHIRULOS	TOON'S	GNOMOS	PRINCIPES	BAM BAM	CELESTINOS	VALU'S	LOS CHAPITAS
2	JADE	DIABLITOS VERDES	COSME Y DAMIAN	DON BARULLO	MANDALA	LA DESCOCADA	LOS DUENDES	OHANA	FENIX	ZODIACO
3	DESCARA2	COLIBRIQUIS	CHERRY'S	HEREDEROS	FRENESI	YOGUIS	DEPORTADOS	LOS TOBY'S	ZERO	S.C.
4	DULCINEA	MANO A MANO	BUBY'S BIS	LOS BUSCADORES	LA ZAFADA	ADRENALINA	WONKA'S	LOS PEPINITOS	QUIJOTES	SAPHIRUS
Prom. por etapa	**4,35**	**5,78**	**5,89**	**4,37**	**5,98**	**5,57**	**4,98**	**4,27**	**7,74**	**5,20**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Modelo 204

1.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	CHERRY'S	FRENESI	YOGUIS	ZERO	COLIBRIQUIS	S.C.	LOS TOBY'S	DESCARA2	HEREDEROS	DEPORTADOS
2	COSME Y DAMIAN	JADE	MANDALA	DON BARULLO	OHANA	LA DESCOCADA	LOS DUENDES	FENIX	ZODIACO	DIABLITOS VERDES
3	GNOMOS	PRINCIPES	VALU'S	LOS CHAPITAS	ZABRITOS	CACHIRULOS	TROYANOS	CELESTINOS	BAM BAM	TOON'S
4	LOS BUSCADORES	MANO A MANO	WONKA'S	ADRENALINA	LOS PEPINITOS	QUIJOTES	SAPHIRUS	LA ZAFADA	BUBY'S BIS	DULCINEA
Prom. por etapa	**5,83**	**4,83**	**4,78**	**5,97**	**5,09**	**5,39**	**5,42**	**5,88**	**5,97**	**4,98**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	VALU'S	ZABRITOS	PRINCIPES	LOS CHAPITAS	CACHIRULOS	TOON'S	BAM BAM	CELESTINOS	TROYANOS
2	COSME Y DAMIAN	MANDALA	JADE	LA DESCOCADA	DON BARULLO	OHANA	LOS DUENDES	ZODIACO	FENIX	DIABLITOS VERDES
3	CHERRY'S	YOGUIS	COLIBRIQUIS	FRENESI	LOS TOBY'S	ZERO	HEREDEROS	S.C.	DEPORTADOS	DESCARA2
4	LOS BUSCADORES	WONKA'S	MANO A MANO	ADRENALINA	QUIJOTES	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LA ZAFADA	DULCINEA
Prom. por etapa	**5,83**	**4,78**	**5,01**	**5,50**	**4,91**	**5,40**	**5,75**	**5,85**	**6,00**	**5,12**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

1.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	COLIBRIQUIS	S.C.	ZERO	DEPORTADOS	DESCARA2	LOS TOBY'S	HEREDEROS	FRENESI	CHERRY'S	YOGUIS
2	COSME Y DAMIAN	DON BARULLO	MANDALA	LOS DUENDES	JADE	FENIX	DIABLITOS VERDES	ZODIACO	OHANA	LA DESCOCADA
3	TROYANOS	VALU'S	CACHIRULOS	TOON'S	GNOMOS	PRINCIPES	CELESTINOS	LOS CHAPITAS	BAM BAM	ZABRITOS
4	LOS BUSCADORES	WONKA'S	MANO A MANO	ADRENALINA	QUIJOTES	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LA ZAFADA	DULCINEA
Prom. por etapa	**5,39**	**5,42**	**5,41**	**5,44**	**5,38**	**5,43**	**5,43**	**5,42**	**5,43**	**5,40**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	VALU'S	TROYANOS	CACHIRULOS	GNOMOS	TOON'S	PRINCIPES	LOS CHAPITAS	CELESTINOS	BAM BAM	ZABRITOS
2	DON BARULLO	COSME Y DAMIAN	MANDALA	JADE	LOS DUENDES	FENIX	ZODIACO	DIABLITOS VERDES	OHANA	LA DESCOCADA
3	S.C.	COLIBRIQUIS	ZERO	DESCARA2	DEPORTADOS	LOS TOBY'S	FRENESI	HEREDEROS	CHERRY'S	YOGUIS
4	WONKA'S	LOS BUSCADORES	MANO A MANO	QUIJOTES	ADRENALINA	LOS PEPINITOS	BUBY'S BIS	SAPHIRUS	LA ZAFADA	DULCINEA
Prom. por etapa	**5,42**	**5,39**	**5,41**	**5,38**	**5,44**	**5,43**	**5,42**	**5,43**	**5,43**	**5,40**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	CHERRY'S	FRENESI	S.C.	ZERO	DEPORTADOS	COLIBRIQUIS	HEREDEROS	DESCARA2	LOS TOBY'S	YOGUIS
2	LOS DUENDES	JADE	MANDALA	COSME Y DAMIAN	OHANA	LA DESCOCADA	DON BARULLO	ZODIACO	FENIX	DIABLITOS VERDES
3	GNOMOS	TOON'S	VALU'S	BAM BAM	ZABRITOS	CACHIRULOS	PRINCIPES	TROYANOS	LOS CHAPITAS	CELESTINOS
4	LOS BUSCADORES	MANO A MANO	WONKA'S	ADRENALINA	LOS PEPINITOS	QUIJOTES	SAPHIRUS	LA ZAFADA	BUBY'S BIS	DULCINEA
Prom. por etapa	**5,78**	**4,78**	**4,43**	**6,90**	**4,29**	**6,04**	**5,63**	**5,47**	**5,18**	**5,64**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	VALU'S	ZABRITOS	TOON'S	BAM BAM	CACHIRULOS	CELESTINOS	LOS CHAPITAS	PRINCIPES	TROYANOS
2	LOS DUENDES	MANDALA	JADE	LA DESCOCADA	DON BARULLO	COSME Y DAMIAN	OHANA	FENIX	ZODIACO	DIABLITOS VERDES
3	CHERRY'S	S.C.	FRENESI	DEPORTADOS	HEREDEROS	ZERO	COLIBRIQUIS	DESCARA2	YOGUIS	LOS TOBY'S
4	LOS BUSCADORES	WONKA'S	MANO A MANO	ADRENALINA	QUIJOTES	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LA ZAFADA	DULCINEA
Prom. por etapa	**5,78**	**4,43**	**4,64**	**5,02**	**6,21**	**6,44**	**5,45**	**5,38**	**5,88**	**4,92**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	S.C.	COLIBRIQUIS	ZERO	DEPORTADOS	HEREDEROS	DESCARA2	FRENESI	YOGUIS	CHERRY'S	LOS TOBY'S
2	ZODIACO	DIABLITOS VERDES	FENIX	LOS DUENDES	DON BARULLO	JADE	MANDALA	LA DESCOCADA	COSME Y DAMIAN	OHANA
3	LOS CHAPITAS	ZABRITOS	VALU'S	BAM BAM	TOON'S	TROYANOS	GNOMOS	PRINCIPES	CACHIRULOS	CELESTINOS
4	WONKA'S	LOS BUSCADORES	MANO A MANO	QUIJOTES	ADRENALINA	LOS PEPINITOS	BUBY'S BIS	SAPHIRUS	LA ZAFADA	DULCINEA
Prom. por etapa	**4,20**	**4,45**	**7,75**	**6,11**	**5,54**	**4,31**	**5,84**	**5,61**	**6,03**	**4,31**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	LOS CHAPITAS	VALU'S	ZABRITOS	BAM BAM	TROYANOS	TOON'S	GNOMOS	CACHIRULOS	PRINCIPES	CELESTINOS
2	ZODIACO	FENIX	DIABLITOS VERDES	LOS DUENDES	JADE	DON BARULLO	MANDALA	COSME Y DAMIAN	LA DESCOCADA	OHANA
3	S.C.	ZERO	COLIBRIQUIS	DEPORTADOS	DESCARA2	HEREDEROS	FRENESI	CHERRY'S	YOGUIS	LOS TOBY'S
4	WONKA'S	MANO A MANO	LOS BUSCADORES	QUIJOTES	LOS PEPINITOS	ADRENALINA	BUBY'S BIS	LA ZAFADA	SAPHIRUS	DULCINEA
Prom. por etapa	**4,20**	**7,75**	**4,45**	**6,11**	**4,31**	**5,54**	**5,84**	**6,03**	**5,61**	**4,31**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

Modelo 201

1.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	LOS TOBY'S	HEREDEROS	COLIBRIQUIS	FRENESI	DEPORTADOS	S.C.	CHERRY'S	YOGUIS	DESCARA2	ZERO
2	ZODIACO	COSME Y DAMIAN	DON BARULLO	MANDALA	LOS DUENDES	LA DESCOCADA	JADE	DIABLITOS VERDES	FENIX	OHANA
3	GNOMOS	TOON'S	TROYANOS	BAM BAM	ZABRITOS	CACHIRULOS	VALU'S	CELESTINOS	PRINCIPES	LOS CHAPITAS
4	WONKA'S	LOS PEPINITOS	LOS BUSCADORES	BUBY'S BIS	SAPHIRUS	QUIJOTES	MANO A MANO	ADRENALINA	LA ZAFADA	DULCINEA
Prom. por etapa	**5,08**	**5,75**	**5,16**	**5,19**	**5,33**	**5,39**	**5,67**	**5,63**	**5,76**	**5,17**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	GNOMOS	TOON'S	ZABRITOS	CACHIRULOS	TROYANOS	BAM BAM	CELESTINOS	PRINCIPES	LOS CHAPITAS	VALU'S
2	DON BARULLO	ZODIACO	LOS DUENDES	COSME Y DAMIAN	LA DESCOCADA	MANDALA	DIABLITOS VERDES	FENIX	OHANA	JADE
3	HEREDEROS	FRENESI	LOS TOBY'S	DEPORTADOS	COLIBRIQUIS	S.C.	YOGUIS	DESCARA2	ZERO	CHERRY'S
4	WONKA'S	LOS PEPINITOS	SAPHIRUS	BUBY'S BIS	LOS BUSCADORES	QUIJOTES	DULCINEA	MANO A MANO	ADRENALINA	LA ZAFADA
Prom. por etapa	**5,73**	**5,62**	**5,01**	**5,54**	**4,92**	**5,09**	**5,64**	**5,81**	**5,16**	**5,63**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

1.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	DEPORTADOS	LOS TOBY'S	DESCARA2	YOGUIS	S.C.	COLIBRIQUIS	FRENESI	ZERO	HEREDEROS	CHERRY'S
2	LOS DUENDES	FENIX	JADE	LA DESCOCADA	DON BARULLO	COSME Y DAMIAN	ZODIACO	MANDALA	DIABLITOS VERDES	OHANA
3	TOON'S	PRINCIPES	GNOMOS	ZABRITOS	VALU'S	TROYANOS	LOS CHAPITAS	CACHIRULOS	CELESTINOS	BAM BAM
4	ADRENALINA	LOS PEPINITOS	QUIJOTES	SAPHIRUS	WONKA'S	LOS BUSCADORES	BUBY'S BIS	MANO A MANO	DULCINEA	LA ZAFADA
Prom. por etapa	**5,44**	**5,43**	**5,38**	**5,41**	**5,42**	**5,39**	**5,42**	**5,41**	**5,42**	**5,43**

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	ZABRITOS	GNOMOS	TOON'S	PRINCIPES	VALU'S	TROYANOS	BAM BAM	CELESTINOS	CACHIRULOS	LOS CHAPITAS
2	LA DESCOCADA	JADE	LOS DUENDES	FENIX	DON BARULLO	COSME Y DAMIAN	OHANA	DIABLITOS VERDES	MANDALA	ZODIACO
3	YOGUIS	DESCARA2	DEPORTADOS	LOS TOBY'S	S.C.	COLIBRIQUIS	CHERRY'S	HEREDEROS	ZERO	FRENESI
4	SAPHIRUS	QUIJOTES	ADRENALINA	LOS PEPINITOS	WONKA'S	LOS BUSCADORES	LA ZAFADA	DULCINEA	MANO A MANO	BUBY'S BIS
Prom. por etapa	**5,41**	**5,38**	**5,44**	**5,43**	**5,42**	**5,39**	**5,43**	**5,42**	**5,41**	**5,42**

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 CHOCO

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	COLIBRIQUIS	FRENESI	YOGUIS	HEREDEROS	DEPORTADOS	DESCARA2	CHERRY'S	S.C.	ZERO	LOS TOBY'S
2	ZODIACO	LOS DUENDES	COSME Y DAMIAN	DON BARULLO	DIABLITOS VERDES	JADE	LA DESCOCADA	FENIX	MANDALA	OHANA
3	TOON'S	LOS CHAPITAS	ZABRITOS	CACHIRULOS	TROYANOS	PRINCIPES	BAM BAM	VALU'S	GNOMOS	CELESTINOS
4	LOS PEPINITOS	WONKA'S	LOS BUSCADORES	QUIJOTES	SAPHIRUS	MANO A MANO	ADRENALINA	BUBY'S BIS	LA ZAFADA	DULCINEA
Prom. por etapa	5,99	4,70	4,68	5,76	5,26	4,28	5,88	6,81	6,46	4,31

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	LOS CHAPITAS	TOON'S	ZABRITOS	CACHIRULOS	TROYANOS	BAM BAM	CELESTINOS	PRINCIPES	GNOMOS	VALU'S
2	DON BARULLO	ZODIACO	LOS DUENDES	COSME Y DAMIAN	LA DESCOCADA	JADE	DIABLITOS VERDES	MANDALA	OHANA	FENIX
3	FRENESI	YOGUIS	COLIBRIQUIS	DEPORTADOS	HEREDEROS	S.C.	LOS TOBY'S	DESCARA2	ZERO	CHERRY'S
4	WONKA'S	LOS PEPINITOS	SAPHIRUS	QUIJOTES	LOS BUSCADORES	BUBY'S BIS	DULCINEA	MANO A MANO	ADRENALINA	LA ZAFADA
Prom. por etapa	4,53	5,69	6,13	5,71	4,39	4,83	4,82	4,37	6,51	7,15

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA

4.1 AMPL/CPLEX

RONDA 1

	1	2	3	4	5	6	7	8	9	10
1	DEPORTADOS	COLIBRIQUIS	ZERO	YOGUIS	DESCARA2	S.C.	HEREDEROS	CHERRY'S	FRENESI	LOS TOBY'S
2	LOS DUENDES	DIABLITOS VERDES	FENIX	LA DESCOCADA	JADE	ZODIACO	DON BARULLO	COSME Y DAMIAN	MANDALA	OHANA
3	BAM BAM	ZABRITOS	VALU'S	PRINCIPES	TROYANOS	LOS CHAPITAS	TOON'S	CACHIRULOS	GNOMOS	CELESTINOS
4	QUIJOTES	LOS BUSCADORES	MANO A MANO	SAPHIRUS	LOS PEPINITOS	WONKA'S	ADRENALINA	LA ZAFADA	BUBY'S BIS	DULCINEA
Prom. por etapa	6,11	4,45	7,75	5,61	4,31	4,20	5,54	6,03	5,84	4,31

RONDA 2

	1	2	3	4	5	6	7	8	9	10
1	ZABRITOS	BAM BAM	TROYANOS	LOS CHAPITAS	VALU'S	PRINCIPES	CACHIRULOS	TOON'S	GNOMOS	CELESTINOS
2	DIABLITOS VERDES	LOS DUENDES	JADE	ZODIACO	FENIX	LA DESCOCADA	COSME Y DAMIAN	DON BARULLO	MANDALA	OHANA
3	COLIBRIQUIS	DEPORTADOS	DESCARA2	S.C.	ZERO	YOGUIS	CHERRY'S	HEREDEROS	FRENESI	LOS TOBY'S
4	LOS BUSCADORES	QUIJOTES	LOS PEPINITOS	WONKA'S	MANO A MANO	SAPHIRUS	LA ZAFADA	ADRENALINA	BUBY'S BIS	DULCINEA
Prom. por etapa	4,45	6,11	4,31	4,20	7,75	5,61	6,03	5,54	5,84	4,31

HUMORISTAS PARODISTAS LUBOLOS
REVISTAS MURGA