



FACULTAD DE
INGENIERÍA
UDELAR



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Modelos Exactos de Optimización en la Gestión de la Pos-cosecha Agropecuaria. Aplicación: Uruguay

Joaquin Torres Yic

Ramiro Claudio Pombo Barca

Programa de Grado en Ingeniería En Computación
Facultad de Ingeniería
Universidad de la República - Universidad UDELAR

Montevideo – Uruguay

Octubre de 2022



FACULTAD DE
INGENIERÍA
UDELAR



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Modelos Exactos de Optimización en la Gestión de la Pos-cosecha Agropecuaria. Aplicación: Uruguay

Joaquin Torres Yic

Ramiro Claudio Pombo Barca

Tesis de Grado presentada al Programa de Grado en Ingeniería En Computación, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Grado en Ingeniería En Computación.

Tutores:

Dr. Ing. Franco Robledo

MSe. Ing. Agustín López de Lacalle

Montevideo – Uruguay

Octubre de 2022

Torres Yic, Joaquin

Pombo Barca, Ramiro Claudio

Modelos Exactos de Optimización en la Gestión de la Pos-cosecha Agropecuaria. Aplicación: Uruguay / Joaquin Torres Yic. - Montevideo: Universidad de la República, Universidad UDELAR, Facultad de Ingeniería, 2022.

IX, 105 p. 29, 7cm.

Tutores:

Franco Robledo

Agustín López de Lacalle

Tesis de Grado – Universidad de la República, Programa en Ingeniería En Computación, 2022.

Referencias bibliográficas: p. 84 – 87.

1. IO, 2. CPLEX, 3. AMPL, 4. Modelo, 5. Optimización. I. Robledo, Franco, López de Lacalle, Agustín, . II. Universidad de la República, Programa de Grado en Ingeniería En Computación.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Ing. Pablo Rebufello

Dr. Ing. Pablo Sartor

Dr. Ing. Cristina Mayr

Montevideo – Uruguay
Octubre de 2022

Agradecimientos

Nos gustaría tomar esta oportunidad para extender un agradecimiento a todos los involucrados en la generación de este documento, entre ellos nos gustaría agradecer a nuestros tutores el Profesor Franco Robledo y el Profesor Agustín López de Lacalle.

A su vez aprovechamos este momento para agradecer a nuestros familiares y amigos por el apoyo incondicional brindado a lo largo de la carrera, a todos los docentes que nos brindaron su asistencia y a nuestros compañeros con los cuales compartimos clases y obligatorios.

Es con gran orgullo que presentamos este como nuestro último documento a la facultad de ingeniería y deseamos que pueda ser utilizado tanto como una investigación aplicable en la realidad y como una herramienta para investigación futuras.

Muchas Gracias

Ramiro Claudio Pombo Barca y Joaquín Torres Yic

Resumen

En este informe correspondiente al trabajo realizado como Proyecto de grado, muestra la construcción de modelos exactos de optimización para resolver el problema de la pos-cosecha agropecuaria con aplicación en Uruguay. Los mismos se proponen determinar la distribución óptima de los productos para minimizar el costo total del transporte sujeto a un conjunto de restricciones. Con este fin se realizaron cuatro modelos matemáticos los cuales fueron desarrollados de forma incremental, los mismos fueron programados y ejecutados utilizando la herramienta AMPL. Sobre el cuatro modelados se realizaron un conjunto de siete pruebas de validación en donde se observó como los resultados fueron los esperados comprobando su correctitud. Además, se realizó un análisis de sensibilidad en donde se evalúa el comportamiento y rendimiento del modelo ante diferentes circunstancias. Luego se realizó un modelo relajado teniendo en cuenta los resultados de las pruebas de sensibilidad sobre el cual también se realizaron casos, pruebas en donde se observó que el modelo relajado tiene un mejor rendimiento que el cuarto modelo, llegando a ejecutar instancias con un conjunto más grande de datos y en menor tiempo de ejecución. Como parte de los experimentos realizados, también se desarrolló un segundo modelo relajado, esta vez modificando las variables de decisión enteras a reales, sobre el cual también se hicieron un conjunto de pruebas. Como conclusión observamos que los modelos realizados sirven para el objetivo planteado y que en caso de que no se cuente con los recursos suficientes, se recomienda la utilización del modelo relajado, ya que posee un rendimiento mejor. Para finalizar se plantean una serie de puntos como trabajo a futuro, entre los que destaca el poder aplicar el modelado con datos de una empresa específica de forma de comprobar de que forma repercute en los costos.

Palabras clave: Pos-cosecha Agropecuaria, Investigación de Operaciones, AMPL, CPLEX, Modelos de Optimización

Tabla de contenidos

1	Introducción	1
1.1	Descripción del problema	1
1.2	Organización del documento	2
2	Fundamentos teóricos	4
2.1	Investigación de operaciones	4
2.2	Modelado	4
2.3	Programación entera	5
2.4	CPLEX	6
2.5	AMPL	7
2.6	Programación lineal	8
2.7	Complejidad Computacional	8
2.8	Ordenes de complejidad	9
2.9	Problemas P	9
2.10	Problemas NP	10
2.11	Problemas NP-Completos	10
2.12	Problemas NP-Hard	11
2.13	Problemas P y NP	11
2.14	Familia de problemas P y NP	11
3	Trabajos relacionados	13
4	Modelado inicial	19
4.1	Introducción	19
4.2	Parámetros	19
4.3	Variables de decisión	20
4.4	Función Objetivo	20

5	Segundo Modelado	22
5.1	Introducción	22
5.1.1	Cambios realizados al modelo anterior	23
5.2	Parámetros	23
5.3	Variables de decisión	24
5.4	Función objetivo	24
6	Tercer Modelado	28
6.1	Introducción	28
6.1.1	Cambios realizados al modelo anterior	29
6.2	Parámetros	29
6.3	Variables de decisión	30
6.4	Función objetivo	30
7	Cuarto Modelado	34
7.1	Introducción	34
7.1.1	Cambios realizados al modelo anterior:	35
7.2	Parámetros	35
7.3	Variables de decisión	36
7.4	Función objetivo	36
8	Metodología de pruebas	40
8.1	Introducción	40
8.2	Validación de correctitud del cuarto modelo	40
8.2.1	Prueba 1	41
8.2.2	Prueba 2	43
8.2.3	Prueba 3	46
8.2.4	Prueba 4	49
8.2.5	Prueba 5	52
8.2.6	Prueba 6	55
8.2.7	Prueba 7	59
8.3	Pruebas de rendimiento	63
8.3.1	Parámetros de generación de datos	63
8.3.2	Hardware Utilizado	64
8.3.3	Test iniciales	64
8.4	Análisis de sensibilidad: cuarto modelo	65
8.4.1	Análisis de Productores	66

8.4.2	Análisis de Plantas	67
8.4.3	Análisis de Consumidores	67
8.4.4	Análisis de Tipo de Grano	68
8.4.5	Análisis de Tipo de Camiones	69
8.4.6	Análisis de Periodos de Tiempo	69
8.4.7	Análisis de Zafras	70
9	Relajaciones	72
9.1	Pruebas sobre el modelo relajado	74
9.1.1	Pruebas aumentando cantidad de Productores y Consumidores	74
9.1.2	Pruebas aumentando cantidad de periodos de tiempo	76
9.1.3	Comparación con el modelo no relajado	77
9.2	Relajación entera	79
9.2.1	Pruebas sobre la relajación entera	79
10	Consideraciones finales	81
10.1	Consideraciones finales	81
10.2	Trabajos a futuro	82
	Referencias bibliográficas	84
	Anexos	88
Anexo 1	Código en AMPL del modelo 4	89
Anexo 2	Código en Node Js para la generación de pruebas	93
2.1	Código del generador de datos	93
2.2	Témplate de datos utilizado por el generador	102
Anexo 3	Datos de pruebas y resultados	105

Capítulo 1

Introducción

1.1. Descripción del problema

La gestión de la pos-cosecha agropecuaria en Uruguay es un rubro que une productores, plantas de acopio de granos y consumidores, parciales o finales. Las plantas de acopio reciben en determinada época del año grandes volúmenes de producto, que deben ser acondicionados y acopiados. Este recibo se realiza en general en períodos muy específicos del año, donde la actividad de recibo es intensa. Asimismo, el retiro de este producto de las plantas se efectúa de acuerdo a necesidades pre-acordadas con el consumidor, donde adquieren un rol fundamental las exportaciones a otros países. En consecuencia, ocurre también que los períodos de retiro de producto de las plantas son específicos y cortos.

La actividad anterior es aún más compleja si se tiene en cuenta que las distancias habituales entre el punto de partida del producto (productores), las plantas de acopio y el consumidor (puertos para exportación, por ejemplo) son sumamente considerables. Esto hace que los costos de transporte del producto tengan un rol valioso a la hora de la toma de decisiones.

Para una empresa de acopio de granos que tiene múltiples plantas distribuidas a lo largo del país, es complejo tomar la decisión de a qué planta destinar cada producto, de forma de minimizar sus costos.

El objetivo es entonces determinar la solución óptima para este problema.

Dada una distribución de locaciones dispuestas geográficamente (plantas de acopio), un conjunto de puntos de partida del producto (ubicaciones de los productores) y un conjunto de destinos del producto (consumidores) se busca determinar la distribución óptima de los productos en el conjunto de plantas que minimice el costo total del transporte (que es proporcional a las distancias recorridas). Hay además una variedad de restricciones que la solución debe respetar, donde los tiempos de entrega y retiro, las capacidades de las plantas y otras situaciones particulares tienen un rol decisivo.

La solución a este problema tiene un impacto directo e inmediato en la gestión de la pos-cosecha agropecuaria en Uruguay, que a lo mejor de nuestro conocimiento, hasta el momento no ha sido explorada en profundidad mediante un enfoque exacto de optimización.

A lo largo de este documento se busca presentar un modelo de programación lineal entera que represente la realidad descrita anteriormente junto con las restricciones impuestas tanto por la realidad a modelar como limitantes referentes al negocio de transporte de grano [1]. La generación del modelo se afrontó de modo iterativo, por lo que a partir del capítulo 4 se mostraran los modelos generados, comenzando por los modelos más simples y avanzando hasta el modelo final. La solución óptima del problema se halló a través del modelado utilizando en lenguaje AMPL junto con la herramienta CPLEX.

1.2. Organización del documento

El presente documento se encuentra organizado en 10 capítulos, en esta sección se describirá brevemente el contenido de cada uno de ellos. En el [capítulo 1](#) se introduce el informe con la descripción del problema a tratar y la organización del documento. En el [capítulo 2](#) se presentan los fundamentos teóricos con algunas definiciones y conceptos básicos relacionados con el problema de estudio y algunas de las herramientas utilizadas para la resolución del mismo. Luego en el [capítulo 3](#) se presenta un resumen de los trabajos relacionados que se tomaron como referencia para el trabajo realizado en este informe. Entre el [capítulo 4](#) y el [capítulo 7](#) inclusive se presentan los modelos matemáticos de forma incremental explicando las distintas partes del mismo y también comentando las diferencias entre cada una de las iteraciones de los modelos. En el

[capítulo 8](#) se presentan el conjunto de pruebas realizadas sobre el modelo final alcanzado de forma de poder validar la correctitud del mismo y analizar su rendimiento. Debido a los resultados obtenidos en las pruebas de rendimiento realizadas, en el [capítulo 9](#) se presenta un modelo relajado junto con una explicación de los cambios realizados y un conjunto de pruebas analizando su rendimiento en comparación al modelo original. Para finalizar en el [capítulo 10](#) se ven las consideraciones finales de este informe en donde se detallan las conclusiones obtenidas y el trabajo a futuro que se podría realizar.

Capítulo 2

Fundamentos teóricos

En esta sección se aborda la tarea de explicar brevemente los conceptos básicos asociados con el problema a afrontar, entre ellos los métodos, conceptos y reglas necesarias para entender las acciones tomadas a lo largo de este documento.

2.1. Investigación de operaciones

La investigación de operaciones, también conocida como teoría de la toma de decisiones, es una disciplina fundamental para múltiples industrias. Consiste en el uso de modelos matemáticos, estadísticos y algoritmos para determinar la distribución de recursos conforme a un set de restricciones [2]. Su objetivo es realizar la toma de decisiones, con la finalidad de mejorar y optimizar el funcionamiento de los procesos. Para este fin se modela la realidad mediante problemas de optimización, donde se definen variables de decisión del problema, una función objetivo a maximizar o minimizar y un conjunto de restricciones que modelan matemáticamente aspectos a satisfacer de la realidad modelada.

2.2. Modelado

Los modelos permiten la comprensión y la resolución de problemas, implican abstracciones que reflejan interacciones relevantes de las entidades del problema. Para establecer un modelo algebraico, a partir de un problema, se deben determinar:

- Parámetros
- Variables de decisión
- Restricciones
- Función objetivo

Los problemas más usuales se basan en buscar la maximización o minimización de una función sujeta a un conjunto de restricciones. Usualmente, un problema de este tipo se representa de la siguiente forma:

$$\begin{aligned} &\text{Maximizar: } f(X) \\ &\text{sujeto a: } r(X) \leq 0 \\ &\quad X \geq 0 \end{aligned}$$

La función f es la llamada función objetivo, X es un vector de variables de decisión. Lo que se busca es encontrar los valores de X para que la función $f(X)$ tome el mayor valor y que a su vez satisfagan las restricciones ($r(X)$) a las cuales están sujetas.

2.3. Programación entera

La programación entera es el método empleado para resolver problemas que tienen variables de decisión enteras. Estos modelos se han considerado sub-modelos de la programación lineal con la característica de tratar con valores enteros. Los creadores e investigadores de esta técnica fueron Wagner (1950) y Manne (1959), quienes desarrollaron varios métodos de solución. Uno de los primeros enfoques de solución al tipo de problemas que plantea la programación entera, fue el de evaluación de cada posible solución, es decir, cada una de las combinaciones de valores enteros para las variables del problema, conduciendo a una solución óptima exacta. A este tipo de resoluciones se les dio el nombre de métodos de fuerza bruta [3].

Por otro lado, los métodos que buscan hacer uso de las formulaciones, las cuales tratan de captar fielmente cada restricción de un problema y una función objetivo referente al problema ha abordado, se los denomina métodos exactos y permiten obtener la solución óptima global a estos problemas sin necesidad

de estudiar todas las soluciones viables [3].

A su vez, se cuenta con otro tipo de métodos conocidos como métodos heurísticos, los cuales intentan obtener una solución factible "muy buena". Si bien no es posible determinar si la solución obtenida mediante estos métodos es una solución óptima global, sí se puede decir que esta solución se encuentra "cerca" de la solución óptima y, por lo tanto, podrá ser aceptable dependiendo del problema que se desee resolver [2]. Una de las grandes ventajas de estos métodos es que permiten una resolución más veloz de un problema en comparación con los métodos exactos, compensando su relativa imprecisión.

Si se requiere que todas las variables sean enteras se habla de programación lineal entera pura; si se necesita que solo algunas de las variables de decisión sean números enteros, se tiene un problema de programación lineal entera mixta [4]. En otro tipo de problemas solo se permite que las variables tomen un valor de cero o de uno; en estos casos se habla de programación lineal entera binaria (digital); si se requiere que solamente algunas de las variables tomen valores de cero o uno, se tiene un problema de programación lineal entera binaria mixta [4].

Para resolver problemas de programación lineal entera, se conocen varios algoritmos y técnicas tales como son: Ralph Gomory , Ramificación y Acotamiento, Enumeración Exhaustiva / Enumeración Explícita, Enumeración Implícita, Aditivo de Egon Balas y Algoritmos heurísticos [2].

2.4. CPLEX

CPLEX se una herramienta desarrollada por IBM que permite solucionar problemas de programación lineal entera, entre otros, mediante un mecanismo de branch and bound [5]. IBM ILOG CPLEX Optimizer resuelve problemas de programación de enteros, problemas de programación lineal muy grandes utilizando variantes primarias o duales del método símplex o el método del punto interior de la barrera, problemas de programación cuadrática convexa y no convexa, y problemas de restricción cuadrática convexa (resuelto a través de programación de cono de segundo orden, o SOCP).

CPLEX Optimizer tiene una capa de modelado denominada Concert que proporciona interfaces para los lenguajes *C++*, *C* y *Java*. Hay una interfaz de lenguaje Python basada en la interfaz C. Por último, se proporciona un ejecutable independiente de Interactive Optimizer para la depuración y otros fines [6].

Se puede acceder a CPLEX Optimizer a través de sistemas de modelado independientes como *AIMMS*, *AMPL*, *GAMS*, *OptimJ* y *TOMLAB*. Además, AMPL proporciona una interfaz para CPLEX CP Optimizer [7].

El IBM ILOG CPLEX Optimization Studio completo consta de CPLEX Optimizer para programación matemática, CP Optimizer para programación de restricciones, el lenguaje de programación de optimización (OPL) y un IDE estrechamente integrado (CPLEX Optimization Studio).

2.5. AMPL

AMPL es una herramienta de modelado de problemas lineales que provee asistencia en todas las partes del ciclo de vida del modelo, desde su creación hasta la resolución [7].

El sistema AMPL admite todo el ciclo de vida del modelado de optimización (formulación, prueba, implementación y mantenimiento) de una manera integrada que promueve un desarrollo rápido y resultados confiables. Mediante el uso de una representación algebraica de alto nivel que describe los modelos de optimización de la misma forma en que la gente piensa acerca de ellos. AMPL puede brindarle la ventaja que necesita para implementar con éxito proyectos de optimización a gran escala [8].

AMPL integra su lenguaje de modelado con un lenguaje de comandos para análisis y depuración, y un lenguaje de secuencias de comandos para manipular datos e implementar estrategias de optimización [7].

2.6. Programación lineal

La programación lineal es un método mediante el cual se optimiza, ya sea maximizando o minimizando, una función objetivo, donde las variables están elevadas a la potencia 1. Esto, tomando en cuenta distintas restricciones dadas [4]. A modo de ejemplo se plantea el siguiente sistema de programación lineal:

$$\begin{aligned} & \text{Maximizar: } c^t X \\ & \text{sujeto a: } AX \leq b \\ & X \geq 0 \end{aligned}$$

Donde A es una matriz constante y b y c son vectores constantes

Este tipo de métodos cuenta exclusivamente de problemas donde la función objetivo y las restricciones son lineales, es decir, que las variables de decisión (X) solo cuentan con el exponente 1. Estos modelos también pueden ser parte de la familia de problemas de programación entera si todas las variables de decisión (X) son enteras, en estos casos estamos hablando de programación lineal entera [4].

2.7. Complejidad Computacional

En informática, la complejidad computacional o simplemente complejidad de un algoritmo es la cantidad de recursos necesarios para ejecutarlo. Se presta especial atención a los requisitos de tiempo y memoria. La complejidad de un problema es la complejidad de los mejores algoritmos que permiten resolver el problema.

El estudio de la complejidad de algoritmos dados explícitamente se denomina análisis de algoritmos, mientras que el estudio de la complejidad de problemas se denomina teoría de la complejidad computacional [9]. Ambas áreas están muy relacionadas, ya que la complejidad de un algoritmo es siempre un límite superior en la complejidad del problema resuelto por este algoritmo. Además, para diseñar algoritmos eficientes, a menudo es fundamental comparar la complejidad de un algoritmo específico con la complejidad del problema a resolver. Adicionalmente, en la mayoría de los casos, lo único que se sabe sobre la complejidad de un problema es que es menor que la complejidad de los algoritmos conocidos más eficientes [9]. Por lo tanto, existe una gran su-

perposición entre el análisis de algoritmos y la teoría de la complejidad.

Como la cantidad de recursos necesarios para ejecutar un algoritmo generalmente varía con el tamaño de la entrada, la complejidad normalmente se expresa como una función $n \rightarrow f(n)$, donde n es el tamaño de la entrada y $f(n)$ es la complejidad del peor de los casos (el máximo de la cantidad de recursos que se necesitan sobre todas las entradas de tamaño n) o la complejidad del caso promedio (el promedio de la cantidad de recursos sobre todas las entradas de tamaño n) [9]. La complejidad del tiempo generalmente se expresa como el número de operaciones elementales requeridas en una entrada de tamaño n , donde se supone que las operaciones elementales toman una cantidad de tiempo constante en una computadora dada y cambian solo por un factor constante cuando se ejecutan en una computadora diferente. La complejidad del espacio generalmente se expresa como la cantidad de memoria requerida por un algoritmo en una entrada de tamaño n [9].

2.8. Ordenes de complejidad

Toda función pertenece a un orden de complejidad ($O(f(n))$) que determina la cantidad de recursos necesarios para resolver la función. En otras palabras, a mayor orden se requerirán una mayor cantidad de recursos para poder resolver el problema [10].

Se dice que una función $g(n)$ es de orden $f(n)$ ($O(f(n))$) si:

$$\lim_{n \rightarrow +\infty} \left(\frac{g(n)}{f(n)} \right) \leq +\infty$$

$n \in \mathbb{Z}$

" $O(f(n))$ está formado por aquellas funciones $g(n)$ que crecen a un ritmo menor o igual que el de $f(n)$ " [10].

2.9. Problemas P

La clase P consta de aquellos problemas que se pueden resolver en tiempo polinomial, es decir, existe un algoritmo que lo puede resolver en tiempo $O(n^k)$

en el peor de los casos, donde k es constante y $n \in \mathbb{Z}$ [10].

Formalmente, un algoritmo es un algoritmo de tiempo polinomial, si existe un polinomio $p(n)$ tal que el algoritmo puede resolver cualquier instancia de tamaño n en un tiempo $O(p(n))$.

Estos problemas se denominan tratables, mientras que otros se denominan intratables o superpolinómicos [9].

2.10. Problemas NP

La clase NP consta de aquellos problemas que son verificables en tiempo polinomial. En otras palabras, se trata de problemas donde se puede aplicar un algoritmo polinómico para comprobar si una posible solución es válida o no [10].

En estos casos se requiere de un método de resolución no determinista consistente en aplicar heurísticos para obtener soluciones hipotéticas que se van desestimando (o aceptando) a ritmo polinómico [10].

Todos los problemas de esta clase se pueden resolver en tiempo exponencial mediante una búsqueda exhaustiva.

2.11. Problemas NP-Completos

Estos problemas se caracterizan por ser los problemas conocidos más complejos de la clase NP [10]. Todos los problemas pertenecientes a la clase NP-Completos pueden ser reducidos en tiempo polinomial a un problema NP. Es decir, que existe una manera de relacionar dos problemas (NP y NP-Completo), de manera que la existencia de un algoritmo que resuelve el primer problema, garantiza la existencia de un algoritmo que resuelve el segundo en un tiempo polinómico [9].

2.12. Problemas NP-Hard

Los problemas NP-Hard (también conocidos como NP-complejos y NP-difícil) son un conjunto de problemas NP los cuales pueden ser reducible a un problema NP-Completo en tiempo polinomial [9].

2.13. Problemas P y NP

Todos los problemas en P se pueden resolver con algoritmos de tiempo polinomial, mientras que todos los problemas en NP - P (los problemas NP que no pertenecen a la clase P) son intratables en tiempo polinomial [10].

Es de gran interés para el campo de Investigación Operativa el saber si $P=NP$. De ser este el caso se podrían resolver todos los problemas NP en tiempo polinomial, lo que significaría un enorme avance en la resolución de problemas complejos en una enorme cantidad de áreas [10].

No se sabe si $P \neq NP$. Sin embargo, se conocen muchos problemas en NP con la propiedad de que si pertenecen a la clase P, entonces se puede probar que $P \in NP$ [9].

2.14. Familia de problemas P y NP

La siguiente figura muestra gráficamente la familia de problemas P y NP en el los casos $P = NP$ y $P \neq NP$.

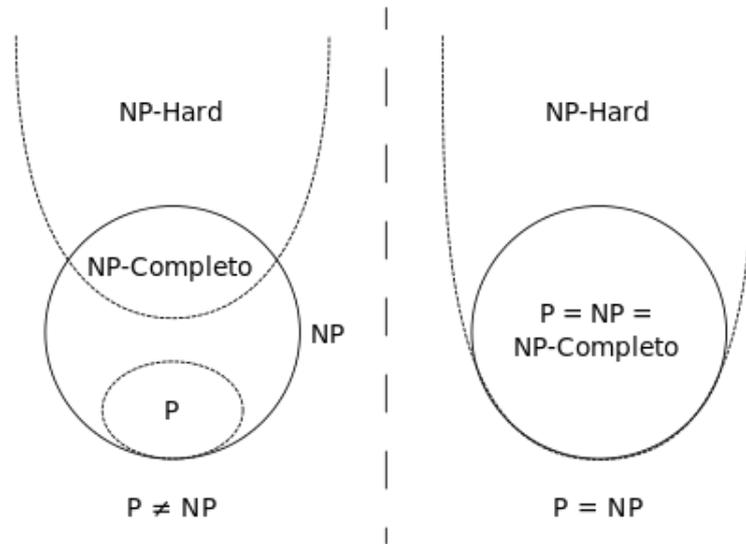


Figura 2.1: Representación de la familia de problemas N y NP [11]

Como se puede apreciar en el diagrama, todos los problemas P son parte de los problemas NP por la propia definición de los problemas NP.

A su vez los problemas NP-Completos también son considerados problemas de clase NP, pero no problemas de clase P. Los problemas NP-Completos también forman parte de los problemas NP-Hard, ya que se pueden reducir en tiempo polinómico a sí mismos, cumpliendo así la definición de un problema NP-Hard [11].

En el caso de que $P = NP$ todos los problemas de la familia NP-Completos también forman parte de la familia de problemas P ($NP-Completo = P$). Esto se debe a que por definición $NP-Completo \subset NP$ y como $NP = P$ entonces $NP-Completo \subseteq P$. Recíprocamente un problema $H \in P$ puede reducirse en tiempo polinomial (aplicando la operación de identidad) a un problema NP (Dado que $I(H) = H$ y $H \in P = NP$) y por tanto $H \in NP-Completo$. Dado que esto se aplica para todo H , $P \subseteq NP$ [10].

Por último es necesario destacar que en este caso todos los problemas forman parte de los problemas NP-Hard dado que $NP-Completo \in NP-Hard$ por definición [10].

Capítulo 3

Trabajos relacionados

En este capítulo se presentan trabajos relevados en la literatura que consideramos similares o cercanos a los abordados en este proyecto tanto en contextos similares como en otros contextos y otros trabajos que han sido de utilidad para el abordaje de la modelización del nuestro problema en forma exacta y su resolución. De cada uno de ellos abordaremos brevemente sus puntos mas importantes que nos han servido de referencia para el mismo.

En el trabajo de Martínez López et al. [12], los autores abordan el problema de distribución de insumos relacionados con el servicio asistencial a pacientes hospitalizados por COVID, en este caso el objetivo es minimizar la distancia total recorrida por un conjunto de vehículos ubicados en un almacén central y que son los encargados de la distribución de los insumos médicos mencionados anteriormente, además se disponen de restricciones adicionales debido a que las unidades de carga no son homogéneas y los distintos tipos de insumos no pueden ser trasladados en cualquier tipo de vehículo . Para resolver dicho problema de enrutamiento de vehículos y asignación de recursos se utilizaron distintos métodos heurísticos y luego se comparan los resultados obtenidos a partir de la utilización de dichos métodos.

Los métodos de resolución heurística usados son:

- EDA (estimation of distribution algorithms).
- Simulated annealing
- VNS (variable neighborhood search)

Si bien dichos métodos pueden no ser los empleados para la resolución final de nuestro problema(dado que nuestro enfoque en el proyecto es la resolución en

forma exacta del problema a partir de un correcto Modelado Exacto del mismo para lograr si es posible la optimalidad global u óptimos locales de buena calidad), nos abre el panorama y nos muestra que tipos de mecanismos existen para tratar de llegar a minimizar un problema de distribución grande sobre el cual no es posible hallar una solución exacta.

Otro trabajo relacionado similar al problema de nuestro caso de estudio se muestra en el trabajo de Macías et al. [13], en el cual los autores introducen el problema de la distribución y transporte de cebada de malta en México a centros de acopio y plantas procesadoras que luego es exportada al exterior. El objetivo del problema planteado en el artículo es determinar la mejor asignación que permita minimizar los costos logísticos asociados con el transporte contando además con restricciones de oferta y demanda. Este caso es útil para poder observar como se plantea un modelo de resolución de este tipo de problemas de minimización, ya que el mismo es muy similar a nuestro caso de estudio, con la diferencia de que en nuestro caso el problema cuenta con un conjunto mayor de restricciones adicionales al suministro de productos alimenticios. Para resolver el problema en este caso se utilizó la herramienta de Solver de Excel y los resultados obtenidos mostraron que la utilización de los métodos de optimización permitieron mejorar la toma de decisiones minimizando el costo de transporte entre granjeros, centros de acopio y malterías.

Si bien en el trabajo [14] los autores no tienen como objetivo principal resolver un caso práctico de optimización como los que mencionamos anteriormente, el mismo es de interés debido a que trata de comparar el rendimiento de diferentes métodos de resolución de problemas de distribución y transporte y de cómo se pueden modelar los mismos, lo cual es una lectura interesante que nos brinda mecanismos que son de utilidad para la realización del modelado matemático de nuestro problema.

En el artículo de Karel Jeřábek y Mária Stopková [15] se repasan diferentes métodos de investigación operativa asociada a tareas de distribución, pero a diferencia de nuestro caso de aplicación están orientadas a la logística de la distribución y optimización de rutas en las ciudades, en particular busca técnicas aplicadas a la resolución de asignación de rutas para vehículos de transporte. Para ello, representan la tarea de carga y traslado en forma de grafos para

luego aplicar técnicas de la teoría de grafos para optimizar las rutas de transporte.

El trabajo de Pečený et al. [16], los autores se focalizan en explicar la necesidad de los métodos de optimización aplicados al transporte y procesos logísticos de hoy en día, permitiendo una reducción en los costos de transporte, procesos de almacenamiento y producción, además del aumento en la eficiencia del tiempo. Para ello se centra en la utilización de método del vecino más próximo [17] y del método de aproximación de Vogel [18] explicando cada uno de los métodos en donde luego los aplica a un caso práctico de un negocio que posee su propia flota de transporte y desea optimizar los costos operativos. Como resultado de la utilización de ambos métodos se pudo observar que con ambos se logra reducir los costos y en este caso en particular el método de aproximación de Vogel fue el que logro ser mas efectivo.

Con respecto al modelado utilizando el lenguaje AMPL nos encontramos con el trabajo de Christian Kirches, Hans Georg Bock y Sven Leyffer [8] en donde se centran en el uso de extensiones de AMPL para la aplicación de problemas de optimización matemática. En el mismo se describen elementos de la sintaxis del lenguaje para modelar de forma conveniente la ecuación diferencial ordinaria (ODE) [19] y la ecuación algebraica diferencial (DAE) [20] restringidas a problemas de control óptimo de enteros mixtos (MIOCPs) [21].

Otro caso de estudio práctico en donde se busca la resolución de un problema de optimización de la cadena logística lo encontramos en el trabajo de Al Theeb et al. [22] en donde se propone un modelo de optimización ejecutado con CPLEX para resolver el problema de selección de rutas para el transporte de productos en camiones refrigerados que no deben romper la cadena de frío para los cuales es clave el poder realizar los traslados por la ruta óptima en el menor tiempo posible para lograr reducir los costos operativos. Como resultado del trabajo de los autores se observa que la utilización del modelado realizado funciona para reducir los costos operativos llegando a ahorrar un 9,25 % de los costos de distribución de dicho caso de estudio.

En el trabajo realizado por Govin et al. [23] los autores se centran en estudiar el problema del vendedor viajero que trata de encontrar el camino más

corto entre un conjunto de clientes y regresar al primero. Para ello los mismos decidieron resolver el problema primero mediante el método heurístico Cuckoo Search Optimization [24] para luego comparar los resultados obtenidos con la resolución del mismo problema esta vez obtenida mediante la realización de un modelado matemático utilizando la herramienta AMPL. Al observar los resultados los mismos mostraron que el modelo realizado en AMPL encontró una solución que requiere recorrer una menor cantidad de millas por lo que resulto ser más efectivo que el método heurístico.

La decisión del uso de la herramienta CPLEX para la resolución del problema abordado queda aún más sólida al estudiar los trabajos de Payman Ghasemi Saghand y Hadi Charkhgard [25], quienes tomaron múltiples problemas de optimización lineal y se dieron a la tarea de resolverlos haciendo uso de múltiples herramientas de resolución. Los resultados de estos experimentos concluyen que la herramienta CPLEX permite solucionar problemas en un tiempo promedio menor a otras herramientas similares tales como Gurobi o Xpress. Cabe mencionar que estos estudios fueron realizados con problemas de programación no lineal y, por tanto, podrían no aplicarse completamente a nuestro caso, sin embargo, el estudio de las herramientas no parece indicar que este haya sido un factor determinante en el rendimiento de las mismas.

Pese a que no está relacionado con la temática de problemas de distribución, el trabajo [26] trata sobre cómo se pueden aplicar mecanismos de investigación operativa en el ámbito de la criminología, en particular el mismo se centra en cómo la investigación operativa se puede utilizar para enfrentarse a la investigación de crímenes asociados al problema de la trata de personas y muestra de que forma sirve para identificar mejores formas para asignar recursos escasos, descubrir patrones en los datos y predecir tendencias y comportamientos futuros.

Relacionado con la mejora de procesos en el sector de la agricultura, el libro de Muzamil et al. [27] trata de las diferentes técnicas de investigación operativa que se pueden aplicar para mejorar distintos aspectos de la ingeniería agrícola como lo son la asignación de recursos, transporte de materiales y ejecución de distintas actividades. En el libro se exploran dichas áreas en donde las técnicas de investigación operativa son aplicables de forma de que el lector pueda ha-

cerse un panorama general sobre las mejoras que se pueden ejecutar en dichos procesos, maximizando así el rendimiento de la cadena productiva.

Para resolver el problema de enrutamiento de vehículos en el trabajo de Baker et al. [28] se propone analizar como se pueden resolver este tipo de problemas mediante la utilización de algoritmos genéticos [29]. En el mismo los autores comentan como los mejores resultados conocidos para este problema se han obtenido mediante búsqueda tabú [30] o Simulated annealing [31]. Las pruebas mostraron como el algoritmo genético puro no alcanza resultados tan óptimos como los métodos mencionados anteriormente pero, por otra parte al hacer un híbrido de un algoritmo genético con métodos de búsqueda de vecindad [32] se puede alcanzar una solución que compite con búsqueda tabú y Simulated annealing en términos de calidad y tiempos de solución.

Uno de los trabajos de optimización, el cual generó un impacto en una industria global, fue realizado en los últimos años por Sun et al. [33], quienes abordaron la tarea de optimizar los horarios de grúas para la industria metalúrgica en el año 2011. El equipo de Sun se enfrentó a un problema similar al que se plantea en este documento, el determinar la forma más eficiente de utilizar un número reducido de grúas de modo de cumplir con la producción de acero con el menor costo posible. La producción del acero consta de múltiples pasos, cada uno con su orden y tiempo definido, para los cuales se puede requerir del uso de una grúa. Por tanto, el lograr determinar la utilización más eficiente posible para estas herramientas conlleva consigo un gran incentivo económico. Al final de los estudios y pruebas ejecutadas en este trabajo se llega a la conclusión de que es posible emplear la herramienta CPLEX para hallar el empleo más eficiente de los elementos necesarios, sin embargo, se nota que este mecanismo deja de ser viable cuando se trata de problemas de gran escala con grandes cantidades de decisiones posibles. A pesar de estos inconvenientes, los resultados obtenidos en este trabajo han tenido un gran impacto en la producción de acero de Shanghái y otras regiones del mundo.

De forma similar, otro de los trabajos que abordan una problemática comparable a la nuestra es el de Vijaya Kumar Manupati y su equipo [34], quienes ante la pandemia por el COVID-19 tomaron el trabajo de hallar la forma más eficiente de distribuir centros de donación de plasma. El plasma luego

podrá ser transportado desde estos centros a los hospitales cercanos vía ambulancia o ser enviados a un banco de plasma vía avión. Si bien a primera vista este problema no parece tener ninguna relación con el problema a abordar en este documento, se pueden trazar claras correlaciones entre el trabajo realizado por Manupati y el de la pos-cosecha. En primer lugar, se nota que ambos problemas tratan con un problema de optimización lineal entera en donde se desea transportar una materia, en un caso plasma y en el otro granos, de una locación inicial hasta una locación final, desde el paciente hasta el hospital o banco de plasma en un caso, y desde el productor hasta un cliente final en el otro, y ambos escenarios requieren que esta materia sea transportada a una locación intermedia para recibir un tratamiento, extracción de plasma en los locales de donación en un caso y tratamiento de granos en las plantas de acopio en el otro. A pesar de estas similitudes es claro notar que se cuenta con dos escenarios diferentes, en el caso de los bancos de plasma el problema se encuentra en donde posicionar los centros de donación, de forma que el costo de transportar el plasma entre estos y los hospitales o el banco sea el menor posible, a su vez debe tenerse en cuenta que estos centros deben estar cerca de los pacientes que realizaran las donaciones. En nuestro caso, la realidad ya se encuentra establecida y, por tanto, no se puede modificar, sin embargo, las estrategias abordadas en el documento [34] serán utilizadas a lo largo de este trabajo para conseguir los mejores resultados posibles.

Otro caso de estudio similar se puede observar en el trabajo realizado por Fatahayu et al. [35] en donde los autores tratan de resolver el problema de selección de rutas de ambulancias transportadoras de bolsas de sangre por parte de los miembros de la cruz roja de Indonesia, ya que se encontró que hay varios hospitales en Jakarta que no reciben las bolsas en el tiempo necesario. Para resolver dicho problema se plantea la utilización de un modelo matemático en AMPL en donde además se compara con una solución alcanzada utilizando algoritmo del vecino mas cercano [17]. Los resultados indicaron que el modelo realizado con el software de AMPL encontró mejores soluciones por aproximadamente 11,52 kilómetros.

Capítulo 4

Modelado inicial

4.1. Introducción

Este capítulo incluye el modelo realizado a partir del relevamiento inicial de los requerimientos del problema que trata de la distribución de la poscosecha agropecuaria, involucrando a los productores, las plantas de acopio y los consumidores, en donde lo que se requiere es buscar minimizar los costos operativos. Los mismos consisten en el costo de contratar camiones para el transporte de granos desde un Productor hasta una Planta de acopio y luego desde esa Planta de acopio hasta un consumidor final. A su vez, el acto de mantener a un camión esperando en una cola de espera generada en la Planta de acopio también conlleva un costo. Para esto se propone el siguiente modelo. Cabe destacar que para esta primera instancia solo se contempla un único tipo de grano.

4.2. Parámetros

I = Cantidad máxima de Productores.

J = Cantidad máxima de Plantas.

K = Cantidad máxima de Consumidores.

$Demanda_k$ = Cantidad de producto que pide el consumidor k .

PrO_i = Cantidad de grano que puede ofrecer la planta i .

$DistPrPl_{ij}$ = Distancia entre el Productor i y la Planta j .

$DistPlCo_{jk}$ = Distancia entre la Planta j y Consumidor k .

$CostoTransp$ = Costo asociado al transporte.

$CostEspera$ = Costo asociado al tiempo de espera.

$Descarga_j$ = Tiempo que se necesita para descargar los granos en Planta j .

4.3. Variables de decisión

X_{ij} = Cantidad que el productor i envía a la planta j .

Y_{jk} = Cantidad que la planta j envía al consumidor k .

E_j = Tiempo de espera en la cola de la planta j (Variable auxiliar).

4.4. Función Objetivo

$$\begin{aligned} \min \quad & \sum_{j=1}^J \sum_{i=1}^I DistPrPl_{ij} X_{ij} \times CostoTransp + \\ & \sum_{k=1}^K \sum_{j=1}^J DistPlCo_{ij} Y_{ij} \times CostoTransp + \\ & \sum_{j=1}^J E_j \times CostEspera \end{aligned}$$

s.a.

1. $\sum_{j=1}^J Y_{jk} = Demanda_k, k = 1, \dots, K.$
2. $\sum_{j=1}^J X_{ij} \leq PrO_i, i = 1, \dots, I.$
3. $\sum_{i=1}^I X_{ij} \times Descarga_j = E_j, j = 1, \dots, J.$

1) Restricción de demanda:

$$\sum_{j=1}^J Y_{jk} = Demanda_k, k = 1, \dots, K.$$

La cantidad que se envía desde las plantas al consumidor k debe ser igual a la demanda del consumidor k .

2) Restricción de oferta:

$$\sum_{j=1}^J X_{ij} \leq PrO_i, i = 1, \dots, I.$$

La cantidad que el productor i envía a las plantas no debe superar a la oferta del productor i .

3) Tiempo de espera en planta:

$$\sum_{i=1}^I X_{ij} \times Descarga_j = E_j, j = 1, \dots, J.$$

El tiempo de espera en la planta j es igual a la cantidad recibida de los productores por el tiempo de descarga en esa planta.

Capítulo 5

Segundo Modelado

5.1. Introducción

Para este modelado se expanden las características del modelado para incluir múltiples tipos de granos. A su vez se modela el tiempo como un parámetro adicional.

Para modelar los distintos tipos de grano se agrega un nuevo subíndice g el cual indicara el tipo de grano enviado. En cuanto al tiempo para el modelado se cuenta con un conjunto de periodos p sobre los cuales se subdividen en un conjunto de tiempos t . Por ejemplo p podría representar las zafras en las cuales se realiza la cosecha y los tiempos t los días correspondientes a la zafra p como se puede observar en la representación gráfica de la *Figura 5.1*.

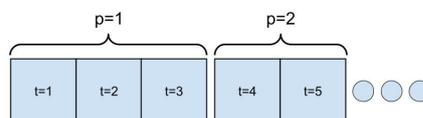


Figura 5.1: Representación de los periodos de tiempo

5.1.1. Cambios realizados al modelo anterior

- Se agregó una nueva dimensión en las variables de decisión X e Y para representar el tipo de grano correspondiente a cada una.
- Se agregó una nueva dimensión en las variables de decisión X e Y para representar los periodos de tiempo en el que se realizan los envíos.
- Se agregó una nueva dimensión en la variable E para representar el tiempo.
- Se agregó una nueva dimensión en los parámetros de *Demanda* (demanda de un consumidor) y *PrO* (Oferta del productor) para representar los periodos de tiempo en el que se realizan los envíos.
- Se agrega el parámetro *VelMed* (velocidad media de los camiones)
- Se agrega el parámetro *InicioPeriodos_[p]* (Para indicar los tiempos que pertenecen a cada periodo)
- Se agregan restricciones y variables referentes a la nueva realidad

5.2. Parámetros

I = Cantidad máxima de Productores.

J = Cantidad máxima de Plantas.

K = Cantidad máxima de Consumidores.

G = Cantidad máxima de Tipos de Grano.

T = Cantidad máxima de periodos de Tiempo para enviar camiones agrupables en periodos de oferta y demanda.

$Demanda_{kgp}$ = Cantidad de producto g que pide el consumidor k en el periodo p .

PrO_{gip} = Cantidad de grano g que puede ofrecer la planta i en el periodo p .

$DistPrPl_{ij}$ = Distancia entre el Productor i y Planta j .

$DistPlCo_{jk}$ = Distancia entre Planta j y Consumidor k .

$CostoTransp$ = Costo asociado al transporte.

$CostEspera$ = Costo asociado al tiempo de espera.

$VelMed$ = Velocidad de cada camión.

$Descarga_j$ = Tiempo de descarga en la planta j .

$InicioPeriodos_p$ = Array que determina los Tiempos (t) pertenecen al Periodo (p).

5.3. Variables de decisión

X_{ijgt} = Cantidad que el productor i envía a la planta j del tipo de grano g en el periodo de tiempo t .

$Y_{jkg t}$ = Cantidad que la planta j envía al consumidor k del tipo de grano g en el periodo de tiempo t .

E_{jt} = Tiempo de espera en la cola de la planta j en el tiempo t (Variable auxiliar).

5.4. Función objetivo

$$\begin{aligned} \min & \sum_{i=1}^I \sum_{j=1}^J \sum_{g=1}^G \sum_{t=1}^T DistPrPl_{ij} X_{ijgt} \times CostoTransp + \\ & \sum_{j=1}^J \sum_{k=1}^K \sum_{g=1}^G \sum_{t=1}^T DistPlCo_{jk} Y_{jkg t} \times CostoTransp + \\ & \sum_{j=1}^J \sum_{t=1}^T \frac{E_{jt} \times CostEspera}{Descarga_j} \end{aligned}$$

S.a.

$$1. \sum_{t=1}^{T_p} Y_{jkgt} = R_{jkgp}, j = 1, \dots, J, k = 1, \dots, K, g = 1, \dots, G.$$

$$\sum_{j=1}^J R_{jkgp} = Demanda_{kpg}, k = 1, \dots, K, g = 1, \dots, G.$$

$$2. \sum_{t=1}^{T_p} X_{ijgt} = Z_{ijgp}, i = 1, \dots, I, j = 1, \dots, J, g = 1, \dots, G.$$

$$\sum_{j=1}^J Z_{ijgp} \leq PrO_{ipg}, i = 1, \dots, I, g = 1, \dots, G.$$

$$3. \sum_{t=1}^{T_p} \sum_{i=1}^I X_{ijgt} = PlO_{jgp}, j = 1, \dots, J, g = 1, \dots, G.$$

$$\sum_{t=1}^{T_p} \sum_{k=1}^K Y_{jkgt} \leq PlO_{jpg}, j = 1, \dots, J, g = 1, \dots, G.$$

$$4. E_{jt} = \sum_{i=1}^I \sum_{g=1}^G X_{ijg(t - \frac{DistPrPl_{ij}}{VelMed})}, j = 1, \dots, J, t = 1, \dots, T.$$

1) Restricción de demanda de los Consumidores en cada periodo:

$$\sum_{t=1}^{T_p} Y_{jkg t} = R_{jk g p}, j = 1, \dots, J, k = 1, \dots, K, g = 1, \dots, G.$$

$$\sum_{j=1}^J R_{jk g p} = Demanda_{k p g}, k = 1, \dots, K, g = 1, \dots, G.$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo (T_p). La suma de todos los camiones enviados desde una planta j a un consumidor k transportando un tipo de grano g debe ser igual a R (variable auxiliar). La suma de todos los envíos realizados por cualquier planta debe ser igual a la demanda exigida por el cliente k para el tipo de grano g en el periodo p .

2) Restricción de oferta de los Productores en cada periodo:

$$\sum_{t=1}^{T_p} X_{ij g t} = Z_{ij g p}, i = 1, \dots, I, j = 1, \dots, J, g = 1, \dots, G.$$

$$\sum_{j=1}^J Z_{ij g p} \leq PrO_{i p g}, i = 1, \dots, I, g = 1, \dots, G.$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo (T_p). La suma de todos los camiones enviados desde un productor i a una planta j transportando un tipo de grano g debe ser igual a Z (variable auxiliar). La suma de todos los envíos realizados por cualquier productor no debe superar a la oferta provista por él para el tipo de grano g en el periodo p .

3) Restricción de oferta de las Plantas en cada periodo:

$$\sum_{t=1}^{T_p} \sum_{i=1}^I X_{ij g t} = PlO_{j g p}, j = 1, \dots, J, g = 1, \dots, G.$$

$$\sum_{t=1}^{T_p} \sum_{k=1}^K Y_{jk g t} \leq PlO_{j g p}, j = 1, \dots, J, g = 1, \dots, G.$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo (T_p). La suma de todos los camiones enviados desde cualquier productor i hacia una planta j de un tipo de grano g en el periodo p se lo denomina $PlO_{j g p}$ (Oferta de Planta). La misma debe ser superior a la suma

de los camiones que salen de la planta j a cualquier cliente k en el periodo p .

4) Tiempo de espera en planta:

$$E_{jt} = \sum_{i=1}^I \sum_{g=1}^G X_{ijg(t - \frac{DistPrPl_{ij}}{VelMed})}, \quad j = 1, \dots, J, t = 1, \dots, T.$$

El tiempo de espera en la planta j en el momento t se calcula como la suma de todos camiones enviados desde cualquier productor hacia la planta j que contengan cualquier tipo de grano y que hayan sido enviados en un momento $T_0 = t - \frac{DistPrPl_{ij}}{VelMed}$ (Tiempo para el cual se calcula de demora menos el tiempo que le tomara al camino llegar a planta).

Capítulo 6

Tercer Modelado

6.1. Introducción

Para este modelado se agregan como restricciones del problema el contar con distintos tipos de camiones. Se asume como hipótesis que los camiones son contratados a un proveedor externo que cuenta con disponibilidad de distintos tipos (los productores no cuentan con flota propia). Además, se cuenta con la restricción de que no todas las plantas pueden aceptar todos los tipos de camiones para emular la realidad.

Para la simplificación del modelado se asume que todos los camiones se mueven a la misma velocidad (promedio) y poseen la misma capacidad de carga (promedio). Esta simplificación se podría levantar agregando un array del largo de cantidad de camiones que represente en cada posición la velocidad a la que se mueve dicho camión y otro array que represente la capacidad de carga, luego habría que hacer las modificaciones pertinentes en las restricciones para tomar en cuenta dichos valores.

Debido a la falta de varianza en los costos de espera en relación con los tipos de camiones la variable que determina el costo de mantener a un camión esperando en una fila no se ve afectada por el tipo de camión.

6.1.1. Cambios realizados al modelo anterior

- Se agregó una nueva dimensión en las variables de decisión X e Y para representar el tipo de camión correspondiente a cada una.
- Se agregó una nueva dimensión en el parámetro $CostoTransp$ (Costo de transporte) para representar el tipo de camión.
- Se agrega el parámetro $PCam$ (Permiso de Camiones en Planta).
- Se agregan restricciones y variables referentes a la nueva realidad

6.2. Parámetros

I = Cantidad máxima de Productores.

J = Cantidad máxima de Plantas.

K = Cantidad máxima de Consumidores.

G = Cantidad máxima de Tipos de Grano.

T = Cantidad máxima de periodos de Tiempo para enviar camiones agrupables en periodos de oferta y demanda.

C = Cantidad máxima de Tipos de Camiones.

P = Cantidad máxima de periodos.(Conjuntos de Tiempo)

$Demanda_{kgp}$ = Cantidad de producto g que pide el consumidor k en el periodo p .

PrO_{gip} = Cantidad de grano g que puede ofrecer la planta i en el periodo p .

$DistPrPl_{ij}$ = Distancia entre el Productor i y Planta j .

$DistPlCo_{jk}$ = Distancia entre Planta j y Consumidor k .

$CostoTransp_c$ = Costo asociado al transporte.

$CostEspera$ = Costo asociado al tiempo de espera.

$VelMed$ = Velocidad de cada camión.

$Descarga_c$ = Velocidad de descarga del tipo de camión c .

$InicioPeriodos_p$ = Array que determina los Tiempos t pertenecen al Periodo p .

$PCam_{jc}$ = Matriz que determina si la Planta j acepta el tipo de Camión c (0 si se acepta, 1 en caso contrario).

6.3. Variables de decisión

X_{ijgtc} = Cantidad que el productor i envía a la planta j del tipo de grano g en el periodo de tiempo t camiones de tipo c .

Y_{jkgtc} = Cantidad que la planta j envía al consumidor k del tipo de grano g en el periodo de tiempo t de camiones de tipo c .

E_{jt} = Tiempo de espera en la cola de la planta j en el tiempo t (Variable auxiliar)

PlO_{jgp} = Cantidad de grano g que puede ofrecer la planta j en el periodo p (Variable auxiliar).

6.4. Función objetivo

$$\begin{aligned} \min \quad & \sum_{i=1}^I \sum_{j=1}^J \sum_{g=1}^G \sum_{t=1}^T \sum_{c=1}^C DistPrPl_{ij} X_{ijgtc} \times CostoTransp_c + \\ & \sum_{j=1}^J \sum_{k=1}^K \sum_{g=1}^G \sum_{t=1}^T \sum_{c=1}^C DistPlCo_{jk} Y_{jkgtc} \times CostoTransp_c + \\ & \sum_{j=1}^J \sum_{t=1}^T \frac{E_{jt} \times CostEspera}{Descarga_c} \end{aligned}$$

s.a.

$$1. \sum_{j=1}^J \sum_{c=1}^C \sum_{t=InicioPeriodos_p}^{InicioPeriodos_p} Y_{jkgtc} = Demanda_{kpg}, \quad k = 1, \dots, K.g = 1, \dots, G.p = 2 \dots P.$$

$$2. \sum_{j=1}^J \sum_{c=1}^C \sum_{t=InicioPeriodos_p}^{InicioPeriodos_p} X_{ijgtc} \leq PrO_{ipg}, \quad i = 1, \dots, I, g = 1, \dots, G, p = 2 \dots P.$$

$$3. \sum_{i=1}^I \sum_{c=1}^C \sum_{t=InicioPeriodos_{p-1}}^{InicioPeriodos_p} X_{ijgtc} = PlO_{jgp}, \quad j = 1, \dots, J, g = 1, \dots, G, p = 2 \dots P.$$

$$\sum_{k=1}^K \sum_{c=1}^C \sum_{t=InicioPeriodos_{p-1}}^{InicioPeriodos_p} Y_{jkgtc} \leq PlO_{jpg}, \quad j = 1, \dots, J, g = 1, \dots, G, p = 2 \dots P.$$

$$4. E_{jt} = \sum_{i=1}^I \sum_{g=1}^G \sum_{c=1}^C X_{ijg(t - \frac{DistPrPl_{ij}}{VelMed})_c}, \quad j = 1, \dots, J, t = 1, \dots, T.$$

$$5. \sum_{c=1}^C \sum_{g=1}^G \sum_{t=1}^T X_{ikgtc} \times PCam_{jc} = 0, \quad j = 1, \dots, J, k = 1, \dots, K.$$

1) Restricción de demanda de los Consumidores en cada periodo:

$$\sum_{j=1}^J \sum_{c=1}^C \sum_{t=InicioPeriodos_{p-1}}^{InicioPeriodos_p} Y_{jkgtc} = Demanda_{kpg}, \quad k = 1, \dots, K.g = 1, \dots, G.p = 2 \dots P$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo ($InicioPeriodos_{p-1}..InicioPeriodos_p$). La suma de todos los envíos realizados por cualquier planta debe ser igual a la demanda exigida por el cliente k para el tipo de grano g en el periodo p .

2) Restricción de oferta de los Productores en cada periodo:

$$\sum_{j=1}^J \sum_{c=1}^C \sum_{t=InicioPeriodos_{p-1}}^{InicioPeriodos_p} X_{ijgtc} \leq PrO_{ipg}, \quad i = 1, \dots, I.g = 1, \dots, G.p = 2 \dots P$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo ($InicioPeriodos_{p-1}..InicioPeriodos_p$). La suma de todos los envíos realizados por cualquier Productor no debe superar a la oferta provista por él para el tipo de grano g en el periodo p .

3) Restricción de oferta de las Plantas en cada periodo:

$$\sum_{i=1}^I \sum_{c=1}^C \sum_{t=InicioPeriodos_{p-1}}^{InicioPeriodos_p} X_{ijgtc} = PlO_{jgp}, \quad j = 1, \dots, J.g = 1, \dots, G.p = 2 \dots P.$$

$$\sum_{k=1}^K \sum_{c=1}^C \sum_{t=InicioPeriodos_{p-1}}^{InicioPeriodos_p} Y_{jkgtc} \leq PlO_{jgp}, \quad j = 1, \dots, J.g = 1, \dots, G.p = 2 \dots P.$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo ($InicioPeriodos_{p-1} \dots InicioPeriodos_p$). La suma de todos los camiones enviados desde cualquier productor i hacia una planta j de un tipo de grano g en el periodo p se lo denomina PlO_{jgp} (Oferta de Planta). La misma debe ser superior a la suma de los camiones que salen de la planta j a cualquier cliente k en el periodo p .

4) Tiempo de espera en planta:

$$E_{jt} = \sum_{i=1}^I \sum_{g=1}^G \sum_{c=1}^C X_{ijg(t - \frac{DistPrPl_{ij}}{VelMed})_c}, \quad j = 1, \dots, J, t = 1, \dots, T.$$

El tiempo de espera en la planta j en el momento t se calcula como la suma de todos camiones enviados desde cualquier productor hacia la planta j que contengan cualquier tipo de grano y que hayan sido enviados en un momento $T_0 = t - \frac{DistPrPl_{ij}}{VelMed}$ (Tiempo para el cual se calcula la demora menos el tiempo que le tomara al camino llegar a planta). En caso de que $T_0 \leq 0 \rightarrow E_{jt} = 0$

5) Resección de tipos de camiones en plantas:

$$\sum_{c=1}^C \sum_{g=1}^G \sum_{t=1}^T X_{ikgtc} \times PCam_{jc} = 0, \quad j = 1, \dots, J, k = 1, \dots, K.$$

La cantidad que envía el productor i a la planta j debe ser transportada en un camión que pueda ser aceptado por la planta j .

Capítulo 7

Cuarto Modelado

7.1. Introducción

Para esta iteración del modelo se considera la existencia de silos ubicados en cada planta. Cada silo puede contener hasta cierta cantidad de grano de un único tipo. Cada planta puede tener un numero diferente de silos.

Debido a la naturaleza de la realidad los silos se encuentran contaminados con un tipo de grano luego de que el mismo se guarda en él, para simplificar el problema vamos a tomar que para cada silo de una planta solo puede contener un cierto tipo de grano como se puede observar en el ejemplo de la *Figura 7.1*.

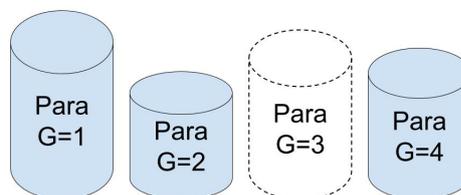


Figura 7.1: Representación de planta que no cuenta con silo para tipo de grano 3

Por ultimo se considera ahora que las plantas requieren de tiempo para pre-limpiar los granos que reciben antes de enviarlos al consumidor final.

7.1.1. Cambios realizados al modelo anterior:

- Se agrega el parámetro *FinPeriodo* (Array de momentos t donde termina un periodo) (Calculado a través de *InicioPeriodo*).
- Se remueve el parámetro *PIO*.
- Se modifica el termino que calcula el tipo de espera.
- Se agregan restricciones y variables referentes a la nueva realidad.

7.2. Parámetros

I = Cantidad máxima de Productores.

J = Cantidad máxima de Plantas.

K = Cantidad máxima de Consumidores.

G = Cantidad máxima de Tipos de Grano.

T = Cantidad máxima de periodos de Tiempo para enviar camiones agrupables en periodos de oferta y demanda.

C = Cantidad máxima de tipos de Camiones.

P = Cantidad máxima de periodos.(Conjuntos de Tiempo).

$Demanda_{kgp}$ = Cantidad de producto g que pide el consumidor k en el periodo p .

PrO_{gip} = Cantidad de grano g que puede ofrecer la planta i en el periodo p .

$DistPrPl_{ij}$ = Distancia entre el Productor i y Planta j .

$DistPlCo_{jk}$ = Distancia entre Planta j y Consumidor k .

$CostoTransp_c$ = Costo asociado al transporte.

$CostEspera$ = Costo asociado al tiempo de espera.

$VelMed$ = Velocidad de cada camión.

$Descarga_c$ = Velocidad de descarga del tipo de camión c .

$InicioPeriodos_p$ = Array que determina los Tiempos t donde comienza cada Periodo p .

$FinPeriodos_p$ = Array que determina los Tiempos t donde termina cada Periodo p .

$PCam_{jc}$ = Matriz que determina si la Planta j acepta el tipo de Camión c . (0 si se acepta, 1 en caso contrario).

$SilosEnPlanta_{jg}$ = Matriz que determina la capacidad de los silos de la Planta j para los tipos de grano g .

$TiempoProcesamiento_j$ = Tiempo que toma la Planta j en procesar la materia prima.

7.3. Variables de decisión

X_{ijgtc} = Cantidad que el productor i envía a la planta j del tipo de grano g en el periodo de tiempo t camiones de tipo c .

Y_{jkgtc} = Cantidad que la planta j envía al consumidor k del tipo de grano g en el periodo de tiempo t de camiones de tipo c .

E_{jt} = Tiempo de espera en la cola de la planta j en el tiempo t (Variable auxiliar).

7.4. Función objetivo

$$\begin{aligned} \min \quad & \sum_{i=1}^I \sum_{j=1}^J \sum_{g=1}^G \sum_{t=1}^T \sum_{c=1}^C DistPrPl_{ij} \times X_{ijgtc} \times CostoTransp + \\ & \sum_{j=1}^J \sum_{k=1}^K \sum_{g=1}^G \sum_{t=1}^T \sum_{c=1}^C DistPlCo_{jk} \times Y_{jkgtc} \times CostoTransp + \\ & \sum_{j=1}^J \sum_{t=1}^T \frac{E_{jt} \times (E_{jt} + 1) \times CostoEspera}{2} \end{aligned}$$

S.a.

$$1. \sum_{j=1}^J \sum_{c=1}^C \sum_{t=1}^{FinPeriodos_p} y_{jkgtc} = \sum_{u=1}^p Demanda_{kgu} , k = 1, \dots, K.g = 1, \dots, G.p = 1 \dots P.$$

(Si el camion no llega a destino dentro del periodo $p \rightarrow y_{jkgtc} = 0$)

$$2. \sum_{c=1}^C \sum_{i=1}^I X_{ijg(\frac{t-DistPrPl_{ij}}{VelMed} - TiempoProduccion_j)_c} \geq \sum_{c=1}^C \sum_{k=1}^K y_{jkgtc}, j = 1, \dots, J.g = 1, \dots, G.t = 1 \dots T.$$

(Si $(\frac{t-DistPrPl_{ij}}{VelMed} - TiempoProduccion_j) \leq 0$ esta restricción se ignora)

$$3. \sum_{i=1}^I \sum_{c=1}^C \sum_{t=InicioPeriodos_p}^{FinPeriodos_p} X_{ijgtc} \leq PrO_{jgp}, j = 1, \dots, J.g = 1, \dots, G.p = 1 \dots P.$$

$$4. E_{jt} = \sum_{i=1}^I \sum_{g=1}^G \sum_{c=1}^C X_{ijg(t - \frac{DistPrPl_{ij}}{VelMed})_c} \times Descarga_c,$$

$$j = 1, \dots, J.t = 1, \dots, T.$$

(Si $(\frac{t-DistPrPl_{ij}}{VelMed}) < 0 \rightarrow E_{jt} = 0$)

$$5. \sum_{g=1}^G \sum_{t=1}^T \sum_{c=1}^C X_{jkgtc} \times AccesoAPlanta_{jc} = 0, i = 1, \dots, I.j = 1, \dots, J.$$

$$6. \sum_{i=1}^I \sum_{k=1}^K \sum_{h=1}^t \sum_{c=1}^C X_{ijghc} - y_{jkghc} \leq SilosEnPlanta_{jg}, j = 1, \dots, J.g = 1, \dots, G.t = 1, \dots, T.$$

1) Restricción de demanda de los Consumidores en cada periodo:

$$\sum_{j=1}^J \sum_{c=1}^C \sum_{t=1}^{FinPeriodos_p} y_{jkgtc} = \sum_{u=1}^p Demanda_{kgu}, k = 1, \dots, K.g = 1, \dots, G.p = 1 \dots P$$

Para cada periodo p , nos aseguramos que cada cliente k reciba en total de todas las plantas j una cantidad del producto igual a la demanda que solicita en el periodo p . A su vez se controla que si el pedido Y salio de la planta j en un periodo p pero llega al cliente en el periodo p' ($p \neq p'$) entonces este no se cuente como entregado en periodo p .

2) Restricción de oferta de las Plantas en cada periodo:

$$\sum_{c=1}^C \sum_{i=1}^I X_{ijg(t - \frac{DistPrPl_{ij}}{VelMed} - TiempoProcesamiento_j)_c} \geq \sum_{c=1}^C \sum_{k=1}^K y_{jkgtc}, j = 1, \dots, J.g = 1, \dots, G.t = 1 \dots T$$

En cada momento de tiempo t se cumple que la cantidad de envíos generados por las plantas j no puede exceder la cantidad total que le llega de los productores i . Solo luego que el envío llega a la planta y fue procesado (en el momento $t_0 = t - \frac{DistPrPl_{ij}}{VelMed} - TiempoProcesamiento_j$) es que se puede enviar desde la planta.

3) Restricción de oferta de los Productores en cada periodo:

$$\sum_{i=1}^I \sum_{c=1}^C \sum_{t=InicioPeriodos_p}^{FinPeriodos_p} X_{ijgtc} = PlO_{jgp}, j = 1, \dots, J.g = 1, \dots, G.p = 1 \dots P.$$

Para cada periodo de tiempo p se tiene el conjunto de momentos t que pertenecen al mismo ($InicioPeriodos_p \dots FinPeriodos_p$). La suma de todos los envíos realizados por cualquier Productor i no debe superar a la oferta provista por

él para el tipo de grano g en el periodo p .

4) Tiempo de espera en planta:

$$E_{jt} = \sum_{i=1}^I \sum_{g=1}^G \sum_{c=1}^C X_{ijg(t-\frac{DistPrPl_{ij}}{VelMed})_c}, j = 1, \dots, J, t = 1, \dots, T.$$

El tiempo de espera en la planta j en el momento t se calcula como la suma de todos camiones enviados desde cualquier productor hacia la planta j que contengan cualquier tipo de grano g y que hayan sido enviados en un momento $T_0 = t - \frac{DistPrPl_{ij}}{VelMed}$ (Tiempo para el cual se calcula la demora menos el tiempo que le tomara al camino llegar a planta). En caso de que $T_0 \leq 0 \rightarrow E_{jt} = 0$

5) Resección de tipos de camiones en plantas:

$$\sum_{c=1}^C \sum_{g=1}^G \sum_{t=1}^T X_{ikgtc} \times PCam_{jc} = 0, j = 1, \dots, J, k = 1, \dots, K.$$

La cantidad que envía el productor i a la planta j debe ser transportada en un camión que pueda ser aceptado por la planta j .

6) Resección de tamaños de silos:

$$\sum_{i=1}^I \sum_{k=1}^K \sum_{h=1}^t \sum_{c=1}^C X_{ijghc} - Y_{jkghe} \leq SilosEnPlanta_{jg}, j = 1, \dots, J, t = 1, \dots, T$$

Para cada instancia de tiempo t la cantidad de granos (de tipo g) acumulados en un silo en una planta j no puede exceder la cantidad de granos que entraron (x) menos la cantidad de granos que salieron (y) hasta ese momento.

Capítulo 8

Metodología de pruebas

8.1. Introducción

Para verificar la correctitud del último modelo se realizaron un conjunto de pruebas orientadas a verificar que este modelo se comporta de la forma esperada bajo cierto conjunto de datos de entrada. Para mostrar ello a continuación comentaremos las pruebas llevadas a cabo junto con los datos obtenidos de las mismas.

8.2. Validación de correctitud del cuarto modelo

En las siguientes pruebas se partió de un caso base y a partir de la modificación del mismo, como por ejemplo apagando una planta, aumentando distancias y modificando ofertas y demandas observamos como la solución cambia respecto a las mismas y así poder observar que el modelo se comporte como es esperado.

Con el fin de simplificar el entendimiento de los siguientes escenarios se plantea lo siguiente:

- El tiempo será representado en Horas.
- El costo total será representado en Dólares.

- A no ser que se especifique, asuma que el costo de transportar una unidad de producto 1 Km es de 1 Dólar.
- A no ser que se especifique, asuma que el costo de esperar a ser atendido en una Planta es de 1 Dólar por hora.
- Asuma que los modelos siempre cuentan con el tiempo suficiente para realizar todos los envíos.
- Todos los modelos cuentan con una única Zafra a no ser que se indique lo contrario.
- Con el fin de simplificar el entendimiento de los resultados de cada prueba la velocidad a la que se desplazan los camiones es de 1 Km/h
- Todas las Plantas cuentan con silos con capacidad suficiente para alojar los granos modelados a no ser que se indique lo contrario.
- El tiempo de producción de todas las plantas es de una hora a no ser que se indique lo contrario.
- Todas las plantas cuentan con acceso para todos los tipos de camiones a no ser que se indique lo contrario.

8.2.1. Prueba 1

Objetivo: Verificar el funcionamiento del modelo a partir de un caso sencillo como se puede ver en la representación de la *Figura 8.1*.

Datos de prueba: Se cuanta con un escenario simple con un único Productor, el cual es capaz de enviar el producto a dos plantas idénticas a distintas distancias.

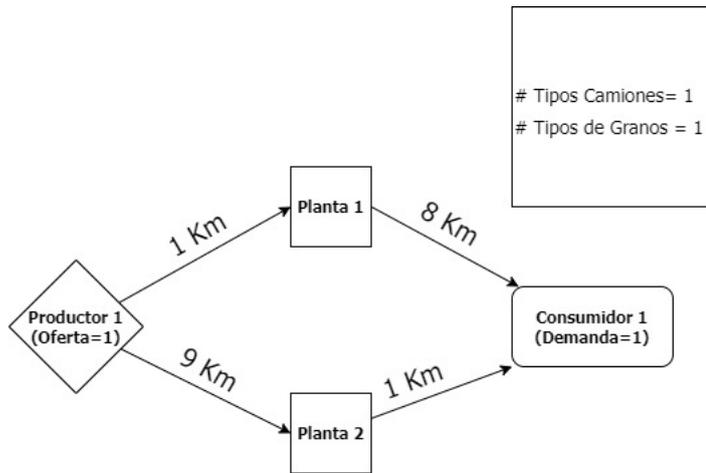


Figura 8.1: Representación gráfica de Prueba 1

Resultados: El modelo eligió correctamente que el Productor envíe por Planta 1 debido a que es la ruta más corta hacia el consumidor final, el valor óptimo encontrado fue de 9.

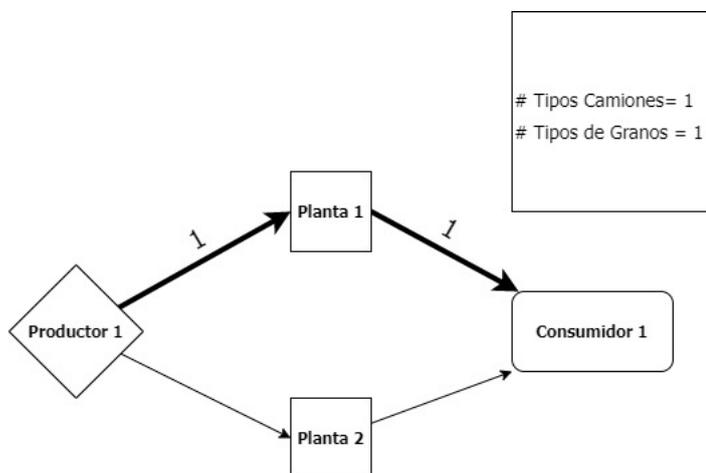


Figura 8.2: Representación gráfica de los Resultados de la Prueba 1

Resultados explícitos:

Envíos realizados desde el Productor:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	1	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde la Planta 1:

Horas	Envíos al Consumidor
1	0
2	0
3	1
4	0
5	0
6	0
7	0
8	0
9	0
10	0

8.2.2. Prueba 2

Objetivo: Verificar la correctitud del modelo al seleccionar el camino más corto entre varias opciones tomando en cuenta el tiempo de producción.

Datos de prueba: Se modificó el ejemplo 1 y se aumentó el tiempo de producción de la Planta 1 de 1 a 3.

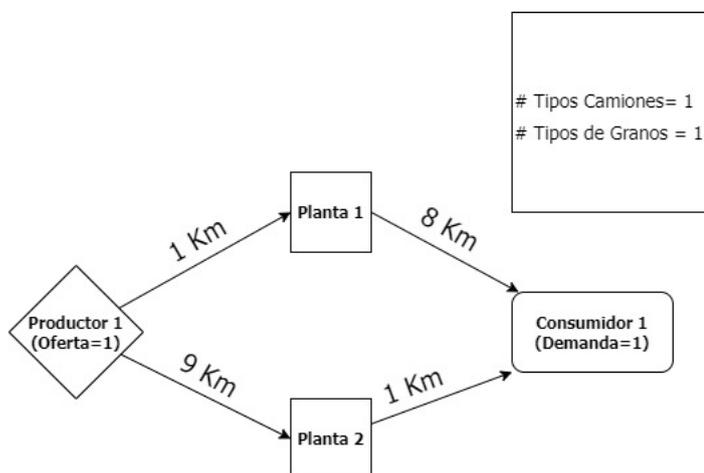


Figura 8.3: Representación gráfica de Prueba 2

Horas que toma procesar granos en planta:

	Tiempo de procesamiento
Planta 1	3
Planta 2	1

Resultados: En este caso el modelo determina correctamente que el costo es menor al enviar el producto por la planta 2 dado que el Costo de espera asociado con la planta 1 es suficiente para justificar el camino más largo. El valor óptimo encontrado fue de 10.

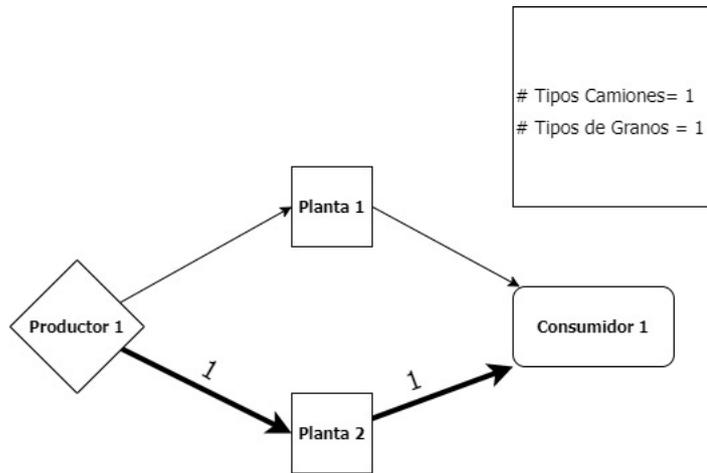


Figura 8.4: Representación gráfica de los Resultados de la Prueba 2

Resultados explícitos:

Envíos realizados desde el Productor:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	0	1
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde la Planta 2:

Horas	Envíos al Consumidor
1	0
2	0
3	0
4	0
5	1
6	0
7	0
8	0
9	0
10	0

8.2.3. Prueba 3

Objetivo: Verificar que se cumpla con la demanda al introducir un nuevo productor.

Datos de prueba: Se modificó el ejemplo 1 agregando a un nuevo productor que tiene oferta 1 y se aumentó en 2 la demanda del consumidor.

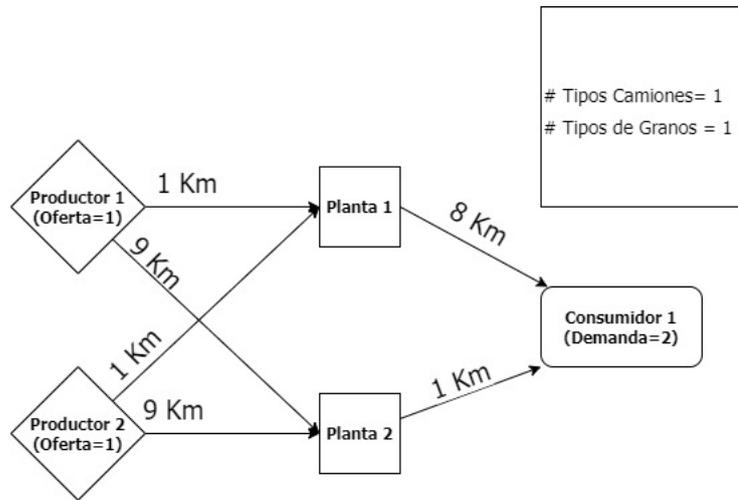


Figura 8.5: Representación gráfica de Prueba 3

Oferta para periodo 1:

	Tipo de grano 1
Productor 1	1
Productor 2	1

Demanda para periodo 1:

	Tipo de grano 1
Consumidor 1	2

Resultados: Ambos Productores determinan correctamente que su camino más corto para cumplir con la demanda es enviar sus productos hacia la Planta 1. El sistema espera un periodo de tiempo entre cada envío para minimizar el costo de espera en la planta.

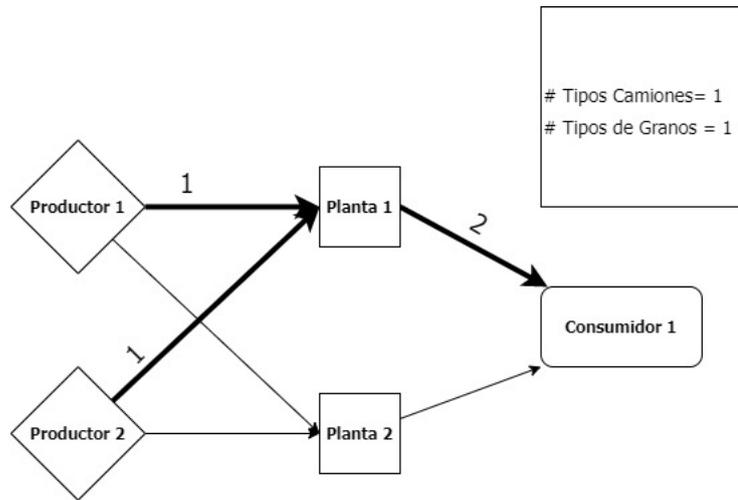


Figura 8.6: Representación gráfica de los Resultados de la Prueba 3

Resultados explícitos:

Envíos realizados desde el Producto 1:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	1	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde el Producto 2:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	0	0
2	0	0
3	1	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde la Planta 1:

Horas	Envíos al Consumidor
1	0
2	0
3	1
4	0
5	1
6	0
7	0
8	0
9	0
10	0

8.2.4. Prueba 4

Objetivo: Introduciendo un nuevo consumidor y tipo de grano, verificar que se cumpla con la demanda.

Datos de prueba: Se modificó el ejemplo 1 agregando a un nuevo consumidor y un nuevo tipo de grano. En este caso, el consumidor 1 demanda 1 unidad del tipo de grano 1 y el consumidor 2 una unidad del tipo de grano 2.

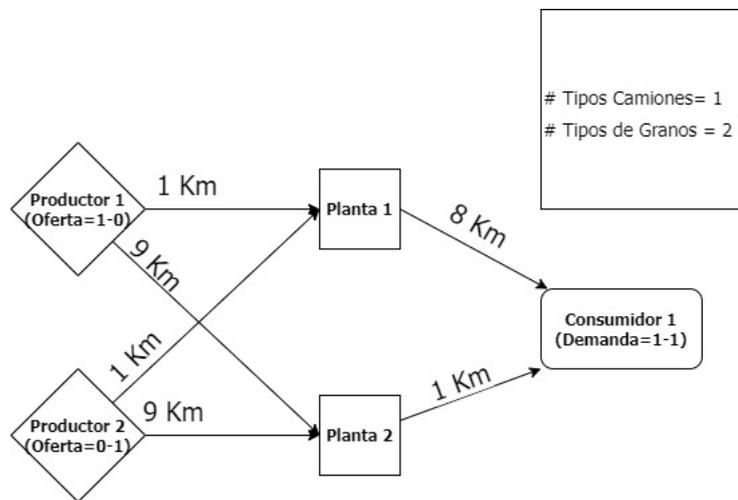


Figura 8.7: Representación gráfica de Prueba 4

Ofertas de los productores:

	Tipo de grano 1	Tipo de grano 2
Productor 1	1	0
Productor 2	0	1

Demanda para periodo 1:

	Tipo de grano 1	Tipo de grano 2
Consumidor 1	1	1

Resultados: Se observa que el productor 1 envía correctamente sus 2 tipos de grano por planta 1, lo cual concuerda con lo esperado, ya que es el camino más corto. Luego de la planta 1 se lleva el tipo de grano 1 al consumidor 1 y el tipo de grano 2 al consumidor 2 cumpliendo así con la demanda.

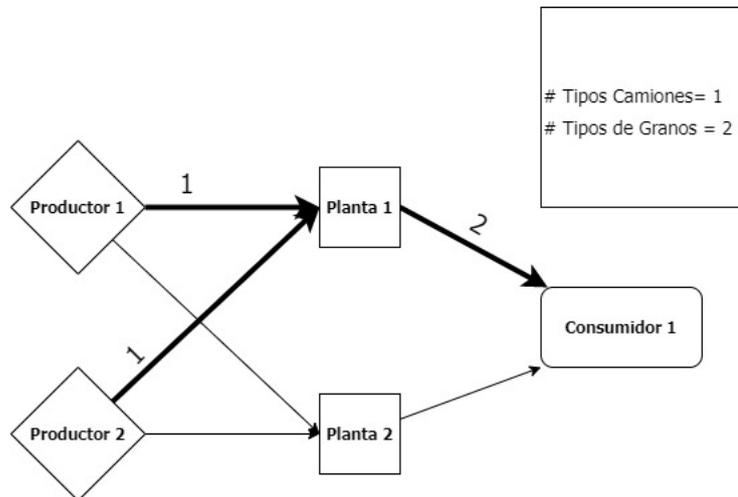


Figura 8.8: Representación gráfica de los Resultados de la Prueba 4

Resultados explícitos:

Envíos realizados desde el Producto 1:

Horas	Planta 1 Grano 1	Planta 2 Grano 1	Planta 1 Grano 2	Planta 2 Grano 2
1	1	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

Envíos realizados desde el Producto 2:

Horas	Planta 1 Grano 1	Planta 2 Grano 1	Planta 1 Grano 2	Planta 2 Grano 2
1	0	0	0	0
2	0	0	0	0
3	0	0	1	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

Envíos realizados desde la Planta 1:

Horas	Envíos al Consumi- dor Grano 1	Envíos al Consumi- dor Grano 2
1	0	0
2	0	0
3	1	0
4	0	0
5	0	1
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

8.2.5. Prueba 5

Objetivo: Verificar que las plantas solo acepten ciertos tipos camiones.

Datos de prueba: Se modificó el ejemplo 1 para que la planta 1 no acepte el tipo de camión 1.

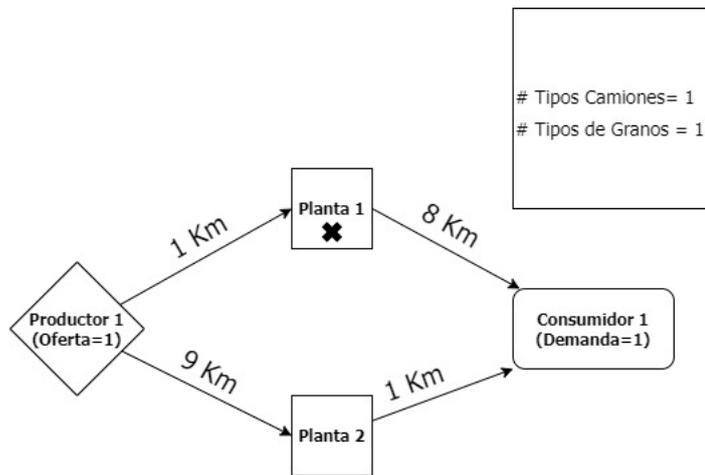


Figura 8.9: Representación gráfica de Prueba 5

Acceso a planta (0 si el camión puede ingresar a la planta):

	Tipo de Camión 1
Planta 1	1
Planta 2	0

Resultados: Según lo esperado el transporte se realiza correctamente por la planta 2 aunque sea el camino más largo dado que la planta 1 no es una opción viable.

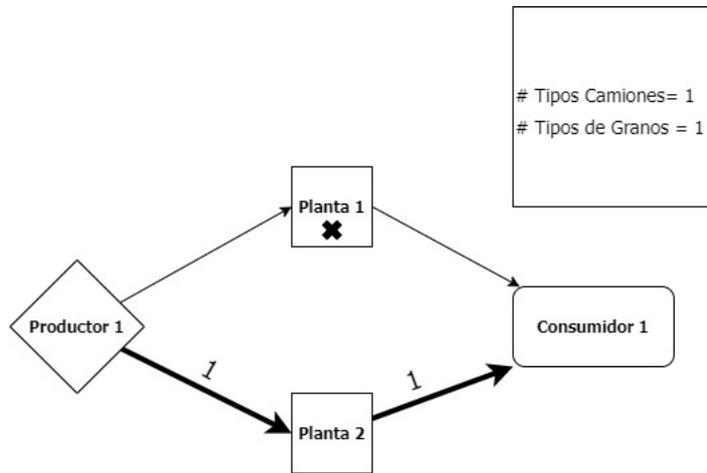


Figura 8.10: Representación gráfica de los Resultados de la Prueba 5

Resultados explícitos:

Envíos realizados desde el Productor:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	0	1
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde la Planta 2:

Horas	Envíos al Consumidor
1	0
2	0
3	1
4	0
5	0
6	0
7	0
8	0
9	0
10	0

8.2.6. Prueba 6

Objetivo: Planteado una segunda Zafra, verificar que se cumpla con la oferta y demanda.

Datos de prueba: Se modifica el ejemplo 1 agregando una nueva zafra, lo que divide nuestras 10 instancias de tiempo en 2. Esto implica la necesidad de especificar la oferta y la demanda del productor y el consumidor.

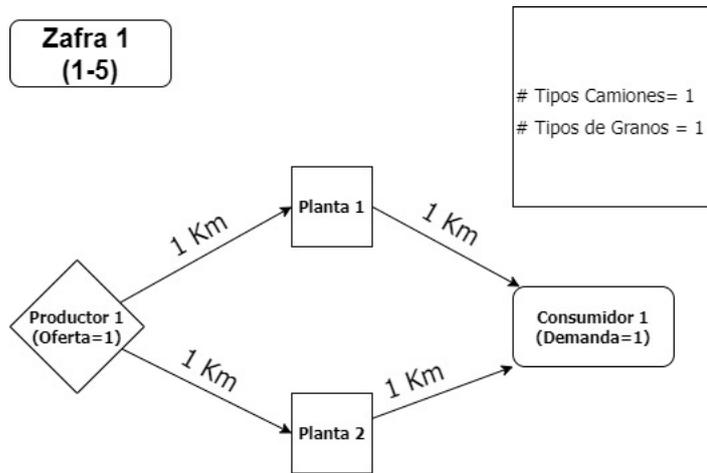


Figura 8.11: Representación gráfica de Prueba 6 Primera Zafra

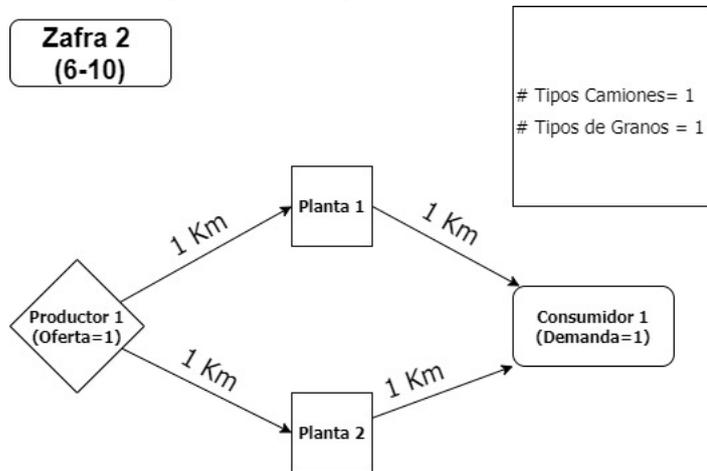


Figura 8.12: Representación gráfica de Prueba 6 Segunda Zafra

Instancias de tiempo a la que pertenecen cada Periodo:

	Tiempos
Primer Periodo	1-5
Segundo Periodo	6-10

Oferta para periodo 1:

	Tipo de grano 1
Productor 1	1

Oferta para periodo 2:

	Tipo de grano 1
Productor 1	1

Demanda para periodo 1:

	Tipo de grano 1
Consumidor 1	1

Demanda para periodo 2:

	Tipo de grano 1
Consumidor 1	1

Resultados: Se envía un camión en cada periodo cumpliendo así correctamente la oferta y la demanda en cada una de las zafras. Dado que ambos caminos cuentan con el mismo consto, el modelo toma el primer camino en la primera zafra y el segundo en la zafra siguiente.

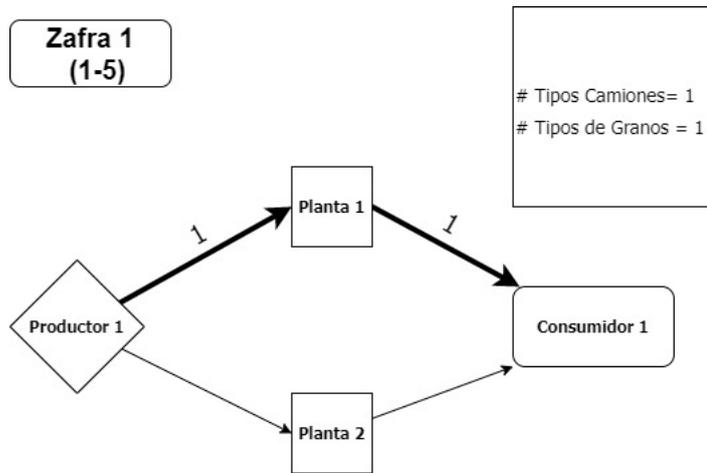


Figura 8.13: Representación gráfica de los Resultados de la Prueba 6 Primera Zafra

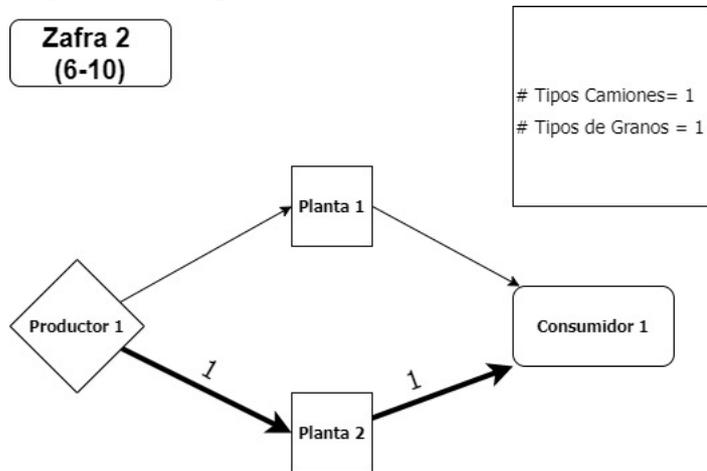


Figura 8.14: Representación gráfica de los Resultados de la Prueba 6 Segunda Zafra

Resultados explícitos:

Envíos realizados desde el Productor durante la primer zafra:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	1	0
2	0	0
3	0	0
4	0	0
5	0	0

Envíos realizados desde la Planta 1 durante la primer zafra:

Horas	Envíos al Consumidor
1	0
2	0
3	1
4	0
5	0

Envíos realizados desde el Productor durante la segunda zafra:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
6	1	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde la Planta 2 durante la primer zafra:

Horas	Envíos al Consumidor
6	0
7	0
8	1
9	0
10	0

8.2.7. Prueba 7

Objetivo: Utilizar la capacidad máxima de los silos para influenciar la cantidad de grano que se envía a cada planta.

Datos de prueba: Se modifica el ejemplo 1 y se cambia la capacidad del silo de la planta 1 a 0, de esta forma sé no será capaz de aceptar más grano.

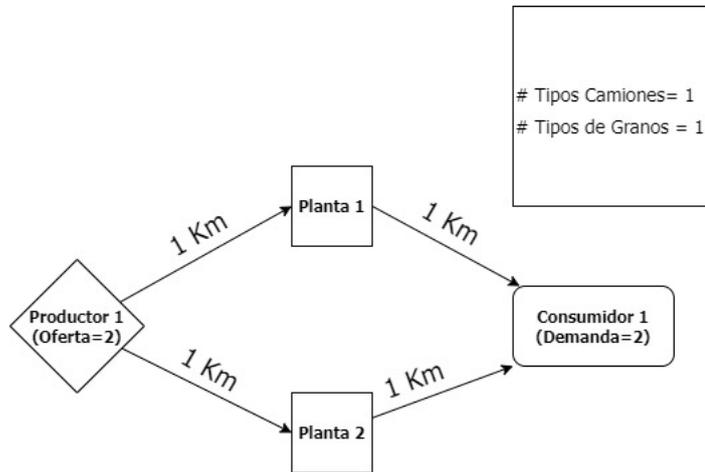


Figura 8.15: Representación gráfica de Prueba 7

Capacidad de silo para un tipo de grano en planta:

	Tipo de Grano 1
Planta 1	0
Planta 2	10

Resultados: Debido a que la planta 1 no puede aceptar más granos (dado que no posee espacio en su silo) el sistema decidió correctamente realizar su envío por la planta 2.

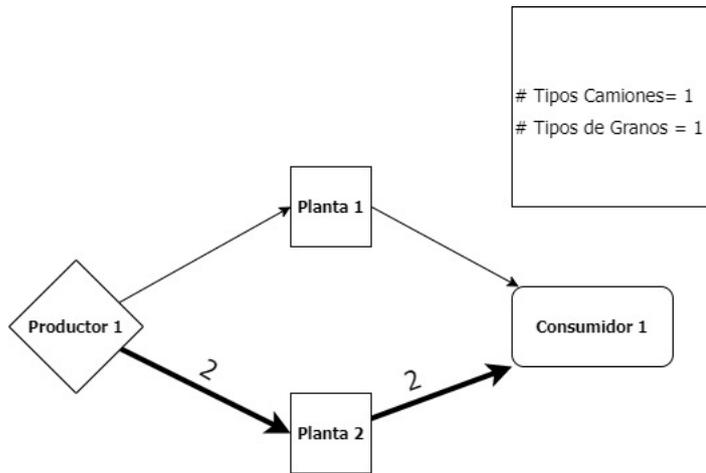


Figura 8.16: Representación gráfica de los Resultados de la Prueba 7

Resultados explícitos:

Envíos realizados desde el Productor:

Horas	Envíos a la Planta 1	Envíos a la Planta 2
1	0	1
2	0	0
3	0	0
4	0	1
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Envíos realizados desde la Planta 2:

Horas	Envíos al Consumidor
1	0
2	0
3	1
4	0
5	1
6	0
7	0
8	0
9	0
10	0

8.3. Pruebas de rendimiento

Luego de las pruebas particulares del comportamiento del modelo vistas en la sección anterior, se realizaron un conjunto de pruebas de rendimiento para evaluar los tiempos de ejecución y comportamiento del modelo al usar un conjunto de datos de entrada similar a la realidad.

Para poder garantizar la creación de pruebas aleatorias y, por tanto, no seleccionadas, especialmente con la intención de obtener resultados favorables, se programó un script que permite generar un conjunto de datos aleatorios de cada uno de los parámetros del modelo dentro de un rango plausible en el caso de aplicación: Uruguay.

8.3.1. Parámetros de generación de datos

Para asegurar un escenario que emule la realidad se toman los siguientes rangos:

Parámetro	Mínimo	Máximo
Productores	100	1000
Plantas	5	15
Consumidores	3	6
Tipos de Granos	50	400
Tipos de Camiones	3	8
Periodos de Tiempo (h)	100	1000
Zafras	1	10
Costo de Transporte (US\$/Km)	1	100
Costo de Espera (US\$/h)	1	20
Oferta (Tn)	10	15000
Demanda (Tn)	3	15000
Distancias entre Productores y Plantas (Km)	1	100
Distancias entre Plantas y Consumidores (Km)	1	100
Capacidad en Silos (Tn)	500	3500
Velocidad de Camiones (Km/h)	60	80

8.3.2. Hardware Utilizado

Todas las pruebas fueron realizadas en el servidor Dantzig de la Facultad de Ingeniería. Los parámetros del servidor son detallados a continuación.

- **CPU** : Intel(R) Core(TM) i9-9900K
- **Numero de cores**: 8
- **RAM**: 65 GB
- **Sistema Operativo**: Rocky Linux 8.4 (Green Obsidian)

8.3.3. Test iniciales

En esta sección se estudiarán los resultados obtenidos luego de ejecutar el modelo con las pruebas generadas utilizando los parámetros descritos en la sección anterior. Debido a la naturaleza aleatoria de los datos de entrada, múltiples simulaciones concluyen de forma instantánea dado la imposibilidad de resolución. En estos casos, el sistema es capaz de determinar esta imposibilidad de resolución en el proceso de pre-solución. Para estos casos también se desea notar que el proceso tarda múltiples horas e incluso días antes de concluir debido a este tipo de errores.

Al ejecutar las pruebas iniciales se encuentran varios inconvenientes referentes a el rendimiento del modelo. El proceso consume una enorme cantidad de recursos, eventualmente causando un error de falta de memoria, lo cual fuerza la terminación de la ejecución. En estos casos también se desea notar la cantidad de tiempo que demora cada uno, algunos logran ejecutar por mas de 24 horas antes de consumir la memoria disponible.

Debido a estas pruebas iniciales, es claro notar que el modelo no está en condiciones de ser utilizado con parámetros de las magnitudes seleccionadas. Con el fin de obtener un modelo que permita simular un ecosistema real, el resto de este documento se enfocara en refinar y optimizar el modelo matemático

y los datos de entrada para reducir los problemas planteados.

8.4. Análisis de sensibilidad: cuarto modelo

En esta sección analizaremos el impacto que tiene cada uno de los parámetros del modelo en los tiempos de ejecución, para ello realizamos un conjunto de pruebas sobre cada uno de los mismos en los cuales aumentamos los valores del parámetro de estudio, medimos los tiempos de ejecución, analizamos la influencia del mismo en el modelo.

Para esta sección no se mencionarán las pruebas de sensibilidad asociadas a los parámetros de Costo de Transporte, Costo de Espera, Oferta, Demanda, Distancias entre Productores y Plantas, Distancias entre Plantas y Consumidores, Capacidad en Silos y Velocidad de Camiones. Debido al rol que cumplen estos parámetros en el modelado, se teoriza que no tienen impacto alguno sobre el rendimiento del modelo y únicamente modificaran el valor óptimo de la función objetivo. Estudios de sensibilidad llevados a cabo posteriormente confirman esta teoría.

Todas las pruebas constarán de los mismos paso a seguir:

- Partiendo de una realidad que contiene:
 - 1 Productor
 - 1 Planta
 - 1 Consumidor
 - 1 Tipo de Grano
 - 1 Tipo de Camión
 - 100 Periodos de Tiempo
 - 1 Zafra
 - Costo de espera igual a 1 por periodo
 - Costo de transporte igual a 1 por unidad de distancia
 - Oferta en cada Productor igual a una unidad por tipo de grano
 - Demanda en cada Consumidor igual a una unidad por tipo de grano

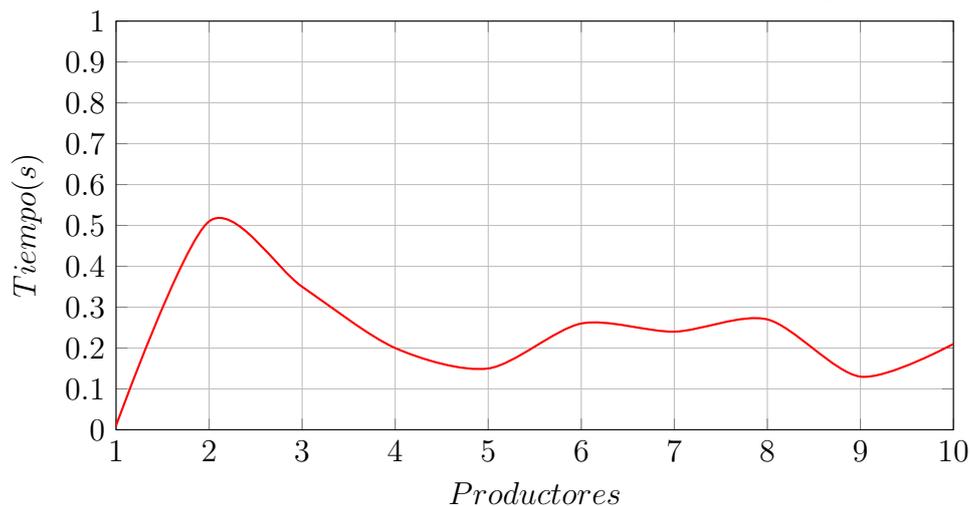
- Distancias entre productores y Plantas igual a 1 unidad de distancia
 - Distancias entre Plantas y Consumidores igual a 1 unidad de distancia
 - Capacidad en Silos igual a 1000 unidades de grano
 - Velocidad de camiones igual a 80 unidades de distancia por periodo de tiempo
- Se selecciona un parámetro para analizar. Manteniendo todos los valores restantes fijos se incrementa el parámetro seleccionado en 1.

Nota: En el caso del análisis de periodos de tiempo, se incrementa el valor en 30 para poder apreciar el impacto de la variable forma más clara.

El objetivo inicial es ejecutar diez iteraciones de estas pruebas para cada parámetro a estudiar. Sin embargo, como se describirá próximamente, esto resulta no ser factible para algunos casos.

8.4.1. Análisis de Productores

Para el caso de los productores se realizaron 10 ejecuciones para analizar el impacto de los mismos, como observamos en la siguiente gráfica:

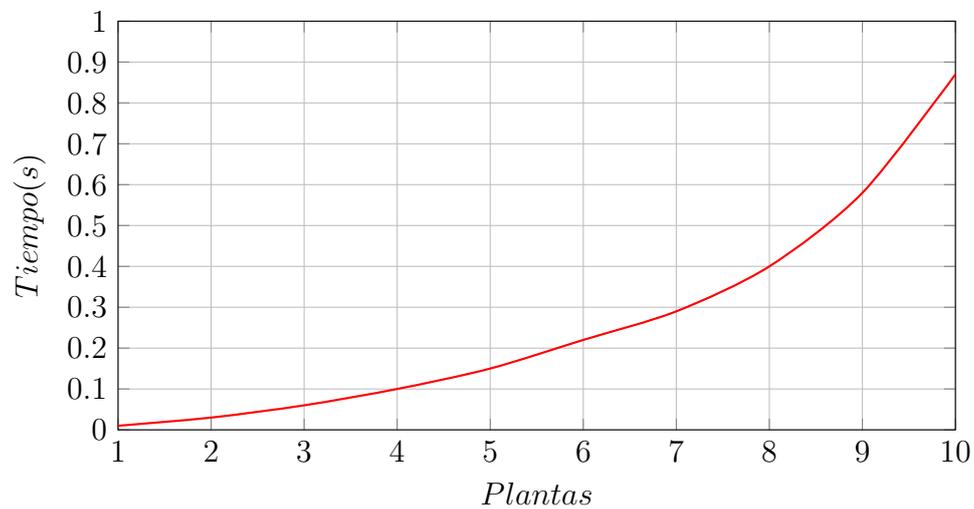


Como se observa, al aumentar la cantidad de productores no se visualiza un impacto grande en los tiempos de ejecución, esto se debe a que en los ejemplos ejecutados al aumentar los productores se aumenta la oferta general del modelo, pero al mantener fijos el resto de los parámetros como la demanda

y consumidores finales el tener oferta sobrante no impacta demasiado en el rendimiento del modelo en parte debido a que no hay una restricción para que toda la oferta sea utilizada.

8.4.2. Análisis de Plantas

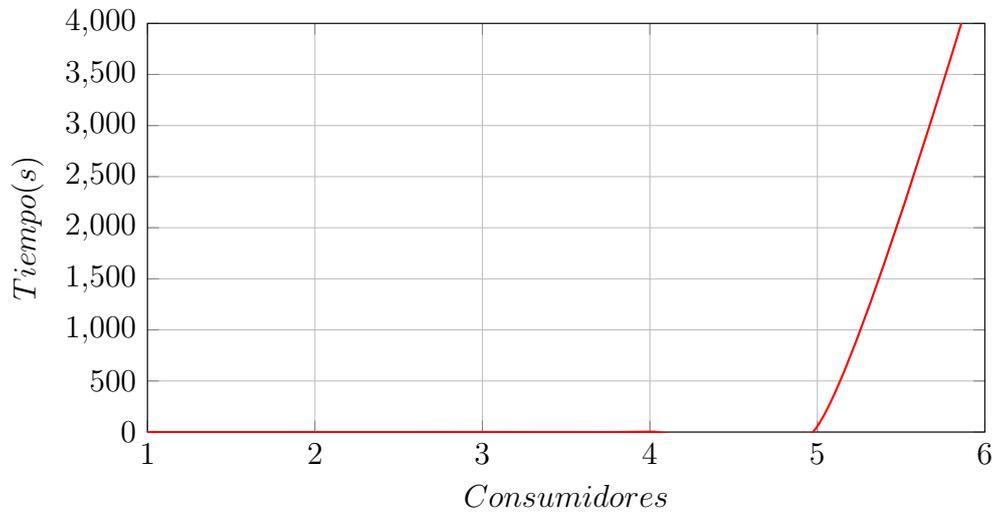
Para el caso de las plantas se realizaron 10 ejecuciones del modelo para evaluar el impacto al aumentar la cantidad de plantas manteniendo el resto de los parámetros fijos.



Como observamos en la gráfica al aumentar la cantidad de plantas aumenta el tiempo de ejecución, y pese a que en segundos no es algo muy notorio en cuanto a la diferencia entre un aumento y el siguiente se observa como el impacto es cada vez mayor. Esto se explica debido a que al aumentar las plantas aumentan la cantidad de decisiones sobre qué caminos tomar, lo cual impacta en el rendimiento del modelo.

8.4.3. Análisis de Consumidores

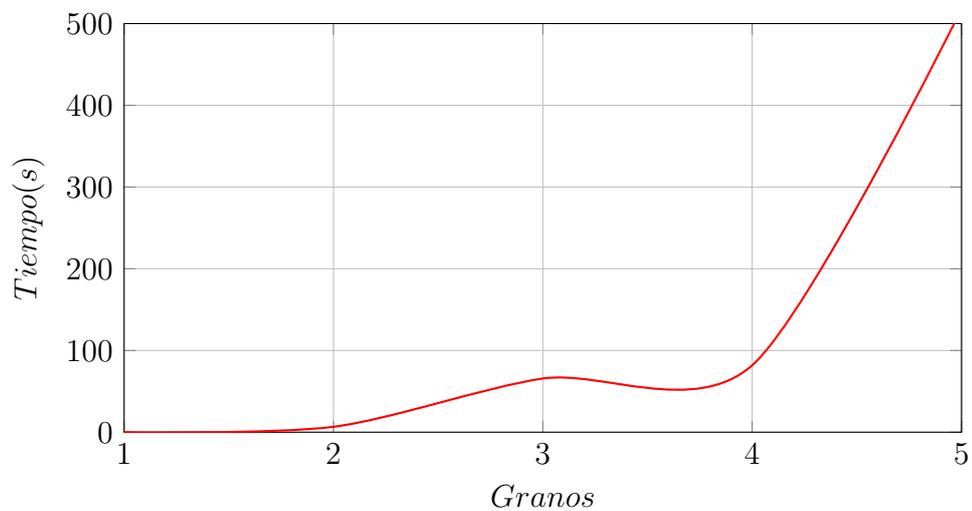
Para esta prueba se aumenta progresivamente la cantidad de Consumidores en el sistema. Para mantener la viabilidad del escenario también se aumenta la cantidad de producto ofrecido, de esta manera se puede cumplir con la nueva demanda. Debido al aumento de tiempo de ejecución se realizan 6 ejecuciones.



Como se puede observar en los resultados, al aumentar la cantidad de consumidores aumenta significativamente el tiempo de ejecución del modelo, esto concuerda con lo esperado, ya que por cada consumidor nuevo implica la asignación de una nueva ruta y todos los recursos que se deben asignar con ello por lo que aumenta la cantidad de decisiones que se pueden tomar.

8.4.4. Análisis de Tipo de Grano

En este análisis se aumenta progresivamente la cantidad de Tipos de Grano en el sistema. Debido al aumento de tiempo de ejecución solo se realizan 5 ejecuciones.

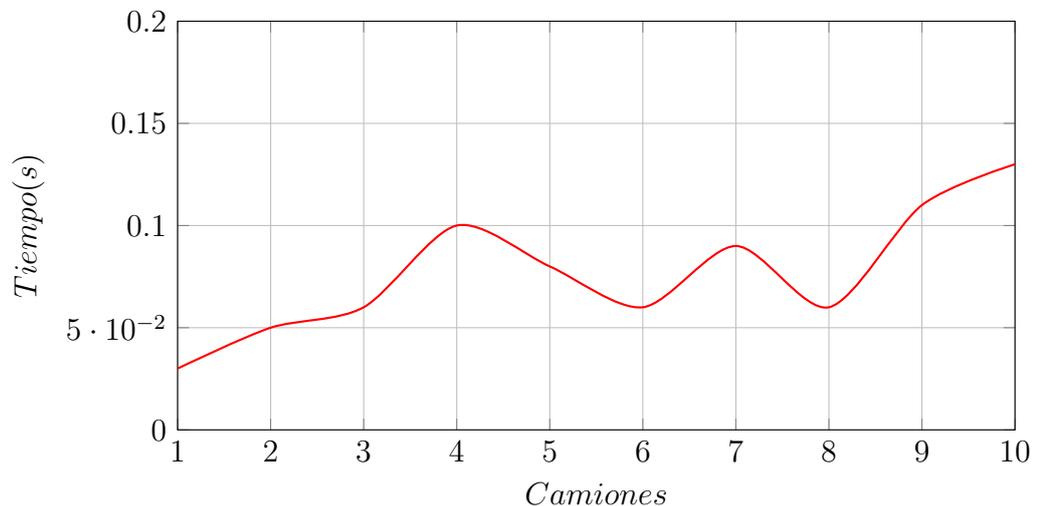


Es claro notar que al aumentar esta variable el tiempo de ejecución del modelo

aumenta de forma considerable. Como se teoriza en el estudio de sensibilidad asociado a la cantidad de Consumidores, el cálculo de nuevos caminos, debido a la distribución de nuevos granos, parece requerir un tiempo de ejecución con crecimiento exponencial.

8.4.5. Análisis de Tipo de Camiones

Para el análisis del parámetro camiones realizamos 10 ejecuciones del modelo aumentando agregando en cada ejecución un nuevo tipo de camión como se observa en la siguiente gráfica.

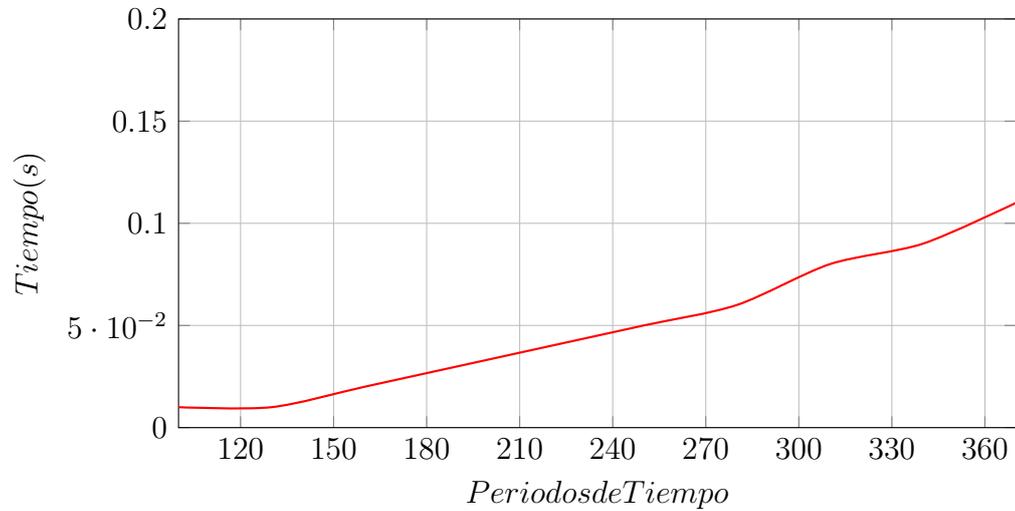


Como observamos al aumentar la cantidad de tipos de camión los tiempos de ejecución varían de forma leve en milésimas de segundo lo que da a entender que las oscilaciones en los tiempos se debe mas a las variaciones que existen entre una ejecución y otra que a la influencia de la cantidad de camiones. Este comportamiento era esperable debido a que la única decisión que impacta es la restricción de los tipos de camiones en planta, por lo cual podemos concluir que este parámetro no tiene gran impacto en el rendimiento del modelo.

8.4.6. Análisis de Periodos de Tiempo

Con el fin de estudiar el impacto que tiene el aumentar los periodos de tiempo en el modelo se realizan 10 ejecuciones en las cuales se aumentan los periodos de tiempo en 30. Se asegura que el modelo tenga resolución válida

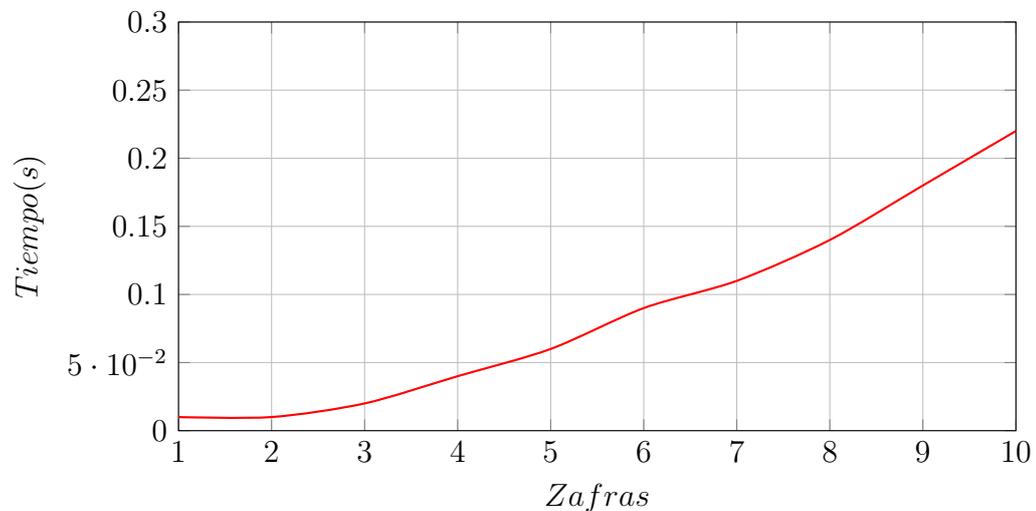
utilizando cualquiera de los valores usados.



Al estudiar esta gráfica se puede notar que el tiempo de ejecución aumenta de forma considerable más allá de que la complejidad del ejemplo se mantiene constante. Es claro notar que la variable Tiempos tiene un impacto relevante sobre el rendimiento del proceso.

8.4.7. Análisis de Zafras

Para analizar la sensibilidad del modelo asociado a la cantidad de zafras modelada se ejecutan 10 pruebas donde se incrementa de forma lineal esta variable. Para mantener la viabilidad de estos modelos también se aumenta la cantidad de periodos de tiempo en 30 al incrementar la variable anterior.



Si bien se nota un incremento el tiempo de ejecución, esto es explicable debido

al aumento de la variable Tiempo. En el caso de que la cantidad de Zafras tenga un impacto en el rendimiento del modelo, este impacto es ínfimo comparado con el impacto generado por la variable Tiempo.

Capítulo 9

Relajaciones

A partir de las diferentes pruebas realizadas en el capítulo anterior se propuso relajar el cuarto modelo de forma de poder ejecutar en tiempos razonables modelos con datos cercanos a lo que nos podemos encontrar en la realidad de la distribución de granos del Uruguay.

El primer punto que observamos es que el aumento en el global de las cantidades repercute fuertemente en la performance, y en especial de las variables que están más ligadas a todas las restricciones y posibilidades que brinda el último modelo presentado. En este caso las que generan más impacto son tanto la cantidad productores, plantas y consumidores, como el aumento de los periodos de tiempo, ya que estas variables son las que pueden aumentar en grandes cantidades (en el Uruguay estamos hablando de entre 100 y 1000 tanto productores como consumidores), y para cada una de ellas hay que evaluar las diferentes alternativas para optimizar los costos. Pese a que dichas variables son las que más impactan en el rendimiento, también son las que tienen menos margen para simplificar, ya que el problema se basa en ellas, por este motivo a ello nos enfocamos en otras variables para poder relajar el modelo y mejorar la performance.

El primer aspecto que notamos para poder relajar el modelo es el impacto de los tipos de granos del mismo. Como se observa en las pruebas, el aumento de la cantidad de tipos de granos tiene un impacto considerable en el rendimiento del modelo, ya que es una dimensión más de la cantidad que se desea optimizar la función objetivo. Debido a ello, decidimos remover los diferentes

tipos de granos para esta relajación y las restricciones asociadas a ellos.

Luego de algunas pruebas, otro aspecto que notamos que influenciaba negativamente en el modelo y se podía simplificar, es el cálculo de los tiempos de espera en planta, por lo que también se decidió removerlo de la función objetivo para el modelo relajado.

A modo de resumen, los cambios en el modelo relajado fueron los siguientes:

- Se removió el parámetro g (Cantidad de tipos de granos).
- Se modificó la función objetivo y el resto de los parámetros para no depender de g .
- Se removió el parámetro $CostoEspera$.
- Se eliminó el cálculo del Costo de Espera en la función objetivo.
- Se eliminó la restricción de tiempo de espera en planta.

$$\min \sum_{i=1}^I \sum_{j=1}^J \sum_{t=1}^T \sum_{c=1}^C DistPrPl_{ij} \times X_{ijtc} \times CostoTransp + \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T \sum_{c=1}^C DistPlCo_{jk} \times Y_{jktc} \times CostoTransp$$

s.a.

$$1. \sum_{j=1}^J \sum_{c=1}^C \sum_{t=1}^{FinPeriodos_p} y_{jktc} = \sum_{u=1}^p Demanda_{kgu}, k = 1, \dots, K.p = 1 \dots P.$$

(Si el camion no llega a destino dentro del periodo $p \rightarrow y_{jktc} = 0$)

$$2. \sum_{c=1}^C \sum_{i=1}^I X_{ij(\frac{t-DistPrPl_{ij}}{VelMed} - TiempoProduccion_j)_c} \geq \sum_{c=1}^C \sum_{k=1}^K y_{jktc}, j = 1, \dots, J.t = 1 \dots T.$$

(Si $(\frac{t-DistPrPl_{ij}}{VelMed} - TiempoProduccion_j) \leq 0$ esta restricción se ignora)

$$3. \sum_{i=1}^I \sum_{c=1}^C \sum_{t=InicioPeriodos_p}^{FinPeriodos_p} X_{ijtc} \leq PrO_{jp}, j = 1, \dots, J.p = 1 \dots P.$$

$$4. \sum_{t=1}^T \sum_{c=1}^C X_{jktc} \times \text{AccesoAPlanta}_{jc} = 0, i = 1, \dots, I, j = 1, \dots, J.$$

$$5. \sum_{i=1}^I \sum_{k=1}^K \sum_{h=1}^t \sum_{c=1}^C X_{ijhc} - y_{jkhc} \leq \text{SilosEnPlanta}_j, j = 1, \dots, J, t = 1, \dots, T.$$

9.1. Pruebas sobre el modelo relajado

Para poder observar el comportamiento del modelo relajado se realizaron una serie de ejecuciones de prueba para evaluar la performance del mismo. Además de ello se realizaron otro conjunto de pruebas que se ejecutaron sobre ambos modelos (el relajado y el original) de forma de poder comparar ambos.

9.1.1. Pruebas aumentando cantidad de Productores y Consumidores

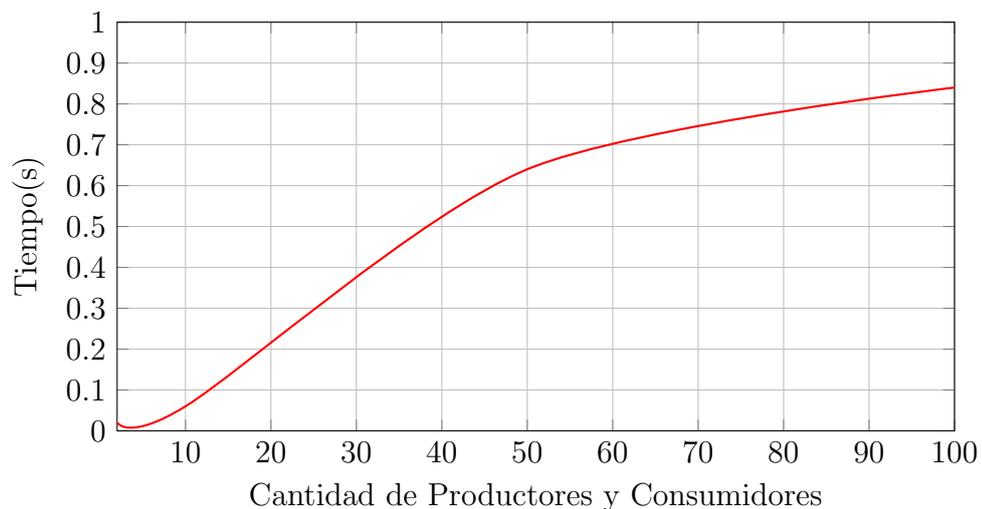
Se ejecutaron pruebas aumentando la cantidad de productores y consumidores. En el modelo original pruebas con más de 10 consumidores y productores suelen terminar con error de memoria. Resulta de interés observar como se comporta el modelo relajado. Para ello llevamos a cabo varias ejecuciones incrementando paulatinamente la cantidad de productores y consumidores.

Para estas pruebas se toma un escenario inicial con:

- 10 Productores
- 4 Plantas
- 10 Consumidores
- 30 Periodos de Tiempo
- 3 Tipos de Camiones
- 2 Zafras
- Demandas en el rango entre 50-100
- Ofertas en el rango entre 100-150
- Distancias entre los Productores y las Plantas en el rango entre 10-100 km

- Distancias entre las Plantas y los Consumidores en el rango entre 10-100 km
- Costo de transporte igual a 11 Dólares por km
- Costo de espera igual a 1 Dólar la hora
- Velocidad de los camiones igual a 80 km/h
- Para facilitar el comportamiento del sistema la Velocidad de Descarga de los camiones es de 1 hora
- La matriz que determina si un Camión es aceptado en una planta se genera de forma aleatoria.
- Para facilitar el comportamiento del sistema, los Silos en cada planta tienen una capacidad máxima de 1000 Ton
- Para facilitar el comportamiento del sistema, cada planta requiere una hora para procesar la materia prima

Para investigar la sensibilidad de este modelo ante un incremento de la cantidad de Productores y Consumidores se aumentan progresivamente, en una unidad, ambas variables de forma simultánea hasta alcanzar un modelo con 100 Productores y 100 Consumidores.



Como se puede observar, se llegó a ejecutar un ejemplo con 100 productores y consumidores finales respectivamente, lo cual muestra que el modelo relajado puede acercarse a la ejecución con conjuntos de datos reales sin inconvenientes.

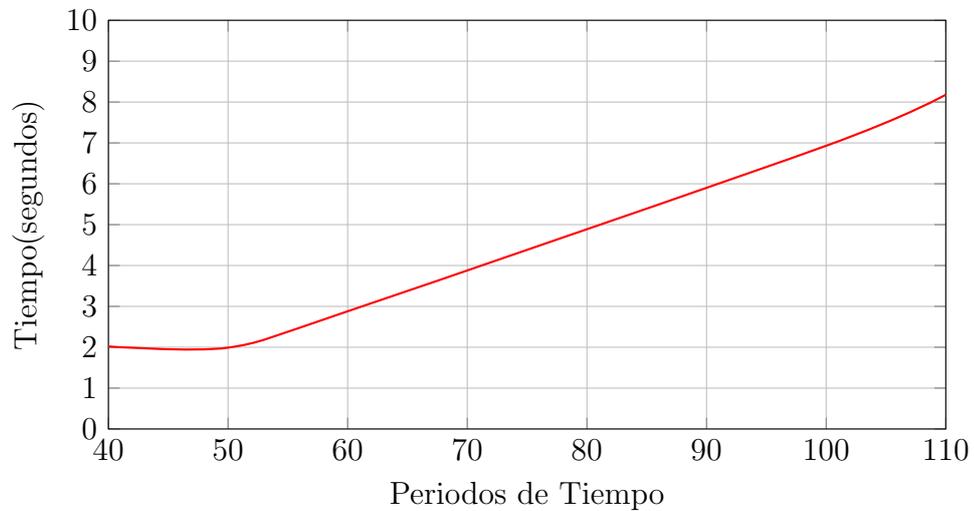
9.1.2. Pruebas aumentando cantidad de periodos de tiempo

Como se pudo observar en pruebas sobre el modelo no relajado, uno de las variables que más impacta sobre la performance del modelo es la cantidad de periodos de tiempo, por ello realizamos varias ejecuciones aumentando la cantidad de periodos de tiempo en cada una de ellas y observamos como se comporta.

Para dichas ejecuciones se utilizó un ejemplo generado con:

- 100 Productores
- 5 Plantas
- 100 Consumidores
- 40 Periodos de Tiempo
- 3 Tipos de Camiones
- 2 Zafras
- Demandas en el rango entre 10-100
- Ofertas en el rango entre 100-150
- Distancias entre los Productores y las Plantas en el rango entre 10-100 km
- Distancias entre las Plantas y los Consumidores en el rango entre 10-100 km
- Costo de transporte igual a 11 Dólares por km
- Costo de espera igual a 1 Dólar la hora
- Velocidad de los camiones igual a 80 km/h
- Para facilitar el comportamiento del sistema la Velocidad de Descarga de los camiones es de 1 hora
- La matriz que determina si un Camión es aceptado en una planta se genera de forma aleatoria.
- Para facilitar el comportamiento del sistema, los Silos en cada planta tienen una capacidad máxima de 1000 Ton
- Para facilitar el comportamiento del sistema, cada planta requiere una hora para procesar la materia prima

Para investigar la sensibilidad de este modelo ante un incremento de la cantidad de Periodos de Tiempo se incrementa gradualmente en 1 el parámetro correspondiente en un rango entre 40-110.



9.1.3. Comparación con el modelo no relajado

Para la comparación con el modelo no relajado se partió de un conjunto de datos base con las siguientes características:

- 2 Productores
- 5 Plantas
- 3 Consumidores
- 30 Periodos de Tiempo
- 3 Tipos de Camiones
- 2 Zafras
- Demandas en el rango entre 10-100
- Ofertas en el rango entre 100-150
- Distancias entre los Productores y las Plantas en el rango entre 10-100 km
- Distancias entre las Plantas y los Consumidores en el rango entre 10-100 km
- Costo de transporte igual a 11 Dólares por km
- Costo de espera igual a 1 Dólar la hora
- Velocidad de los camiones igual a 80 km/h
- Velocidad de Descarga de los camiones en el rango entre 0.1-1 horas
- La matriz que determina si un Camión es aceptado en una planta se genera de forma aleatoria.

- Los silos en cada planta tienen una capacidad máxima en el rango entre 100-1000 Ton
- Cada planta requiere entre 1 y 5 horas para procesar la materia prima

9.1.3.1. Prueba 1

Dicha prueba se realizó con los parámetros mencionados anteriormente sobre ambos modelos, cabe destacar que en el modelo relajado se adecuó la oferta y la demanda para que sea la suma de la oferta y demanda por tipo de granos del modelo original de forma de mantener la oferta y demanda total.

Comparativa de modelados:

Modelo	Tiempo de ejecución(s)	Valor óptimo	GAP
No relajado	0.11	125737	0.00 %
Relajado	0.02	124542	0.00 %

Como se puede observar ambos modelos culminan de forma exitosa encontrando un valor óptimo, en este caso la diferencia de valores óptimos hallados de la función objetivo se puede justificar debido a que el modelo relajado no tiene en cuenta los costos de espera en planta, por lo que es razonable que el valor hallado sea menor. Además, pese a que ambos modelos se ejecutaron de forma casi instantánea, se puede observar como el modelo relajado finaliza su ejecución de forma más rápida.

9.1.3.2. Prueba 2

Para la segunda prueba se decidió aumentar la cantidad de consumidores y productores en diez para observar como impacta el cambio en los respectivos modelos.

Comparativa de modelados:

Modelo	Tiempo de ejecución(s)	Valor óptimo	GAP
No relajado	0.74	372471	0.00 %
Relajado	0.06	366707	0.00 %

Se puede observar que pese a que en ambos el modelo culminó de forma exitosa, alcanzando un valor óptimo, se nota el salto en los tiempos de ejecución entre un modelo y otro con respecto a la prueba 1.

9.1.3.3. Prueba 3

Para la tercera prueba se decidió aumentar la cantidad de periodos de tiempo en 100 y observar como se comportan ambos modelos.

Comparativa de modelados:

Modelo	Tiempo de ejecución(s)	Valor óptimo	GAP
No relajado	8867.38	369345	0.01 %
Relajado	0.59	366707	0.00 %

Cabe aclarar que el modelo original tuvo que ser interrumpido por problemas de recursos, pero se puede observar que impacto tuvo el aumento de los periodos de tiempo en el modelo original y como el modelo relajado culmina de forma rápida, mientras que el modelo original corrió por tiempo prolongado sin poder culminar su ejecución.

9.2. Relajación entera

Como parte de los experimentos realizados, se decidió probar relajando las variables de decisión X e Y que representan las cantidades de producto enviado, quitando la restricción de que dichos valores deben ser enteros para observar como ello repercute en el rendimiento del modelo en comparación con el modelo original.

9.2.1. Pruebas sobre la relajación entera

9.2.1.1. Primera prueba

Para la primera prueba vamos a efectuar una ejecución con un conjunto de datos no demasiado exigentes para ver como se comporta en comparación a los modelos mencionados anteriormente.

Comparativa de modelados:

Modelo	Tiempo de ejecución(s)	Valor óptimo
No relajado	0.11	125737
Relajado	0.59	366707
Relajación entera	0.12	125730.9741

Como observamos en la tabla, el tiempo de ejecución no varió demasiado con respecto al modelo original, pero sí disminuyó el valor objetivo al poder representar los valores con decimales.

9.2.1.2. Segunda prueba

Para la segunda prueba se utiliza el escenario que ejecutamos en la sección anterior, aumentando la cantidad de periodos de tiempo a 100 para ver si el modelo finaliza su ejecución.

Comparativa de modelados:

Modelo	Tiempo de ejecución(s)	Valor óptimo
No relajado	8867.38	369345
Relajado	0.59	366707
Relajación entera	10.53	369296.7336

Analizando los resultados podemos ver que al aumentar considerablemente los periodos de tiempo, el modelo relajado en el cual las variables de decisión X e Y ahora pueden tomar valores reales positivos, finaliza en un tiempo realista para para los tamaños manejados (menos de un minuto), aunque su rendimiento es inferior al modelo relajado planteado inicialmente.

Capítulo 10

Consideraciones finales

10.1. Consideraciones finales

En este documento se logró el objetivo de realizar un modelo el cual presente la realidad a un nivel de detalle satisfactorio. Se comprobaron los resultados de ese modelo y se plantearon optimizaciones para el mismo.

Luego de realizar las pruebas descritas anteriormente, se puede notar que debido al nivel de complejidad del modelo y los recursos utilizados, el problema no logra producir una respuesta satisfactoria en una cantidad de tiempo razonable.

Teniendo en cuenta que el ambiente de pruebas consta de una cantidad de cómputo razonable para este tipo de problemas, se puede concluir que la raíz de estos inconvenientes a nivel de ejecución se encuentra en la complejidad del modelo planteado.

Si bien la representación de restricciones y función objetivo logran funcionar de forma correcta en pruebas controladas, probando así su funcionamiento adecuado, es claro notar que no se pudiera aplicar a casos reales en su estado actual.

Teniendo en cuenta lo planteado anteriormente, se plantean las siguientes alternativas en el caso de que se quiera aplicar este modelo en la realidad. Si se desea emplear el modelo en su estado original para obtener un resultado

exacto, se recomienda contar con las cantidades de recursos necesarios para soportar la ejecución, en particular se debería poseer un espacio de cómputo con una cantidad de memoria RAM superior a los 64 Gb.

Alternativamente, en el caso de que no se cuente con los recursos necesarios, pero aun se desee aplicar el modelo original, es posible ejecutar el mismo reduciendo la cantidad de periodos de tiempo simulados. De este modo, la cantidad de iteraciones del sistema se mantienen acotadas. Al hacer uso del modelo de esta forma se prevendrá que el sistema determine caminos entre productores, plantas y consumidores como válidos debido a la falta de tiempo asociado al sistema. Este tipo de uso puede resultar en una solución la cual difiere del verdadero valor óptimo. En caso de que se desee obtener una solución más cercana a la solución óptima global es necesario repetir la ejecución del modelo con una mayor cantidad de periodos de tiempo, esto brindará un resultado más cercano al óptimo, pero también implicará un incremento en tiempo de ejecución. Quedará a criterio del ejecutor determinar el balance entre rendimiento y exactitud del modelo.

Se desea resaltar que el modelo relajado es capaz de simular casos semejantes a la realidad en Uruguay. Si bien se removieron algunas restricciones, no es menor notar las aplicaciones prácticas que este modelo puede tener. Con esto en mente se recomienda, caso de que sea aplicable, hacer uso del modelo relajado, el cual retorna el mejor resultado posible con la mejor aproximación a la realidad.

10.2. Trabajos a futuro

- Simplificar las restricciones del modelo para generar un problema más simple.
- Conseguir un ecosistema de pruebas con mayor capacidad de cómputo para poder ejecutar las pruebas deseadas.
- En caso de procurar un nuevo ecosistema de pruebas, se podrán agregar nuevas restricciones para simular de forma más completa la realidad.
- A modo de ejemplo, se pueden considerar nuevas acciones a tomar por las plantas de acopio, tales como almacenar los granos entrantes sin pro-

cesarlos.

- Relacionarnos con una empresa de transporte de granos para obtener datos de la realidad y validar que el uso de este modelo reduce los costos operativos del proceso.
- Utilizar distintos algoritmos de resolución en búsqueda de uno que realice las operaciones de forma más eficiente.

Referencias bibliográficas

- [1] “Teske Transporte”. En: (2022). URL: <https://www.teske.com.uy/>.
- [2] Lieberman Frederick S. Hillier - Gerald J. *INTRODUCCIÓN A LA INVESTIGACIÓN DE OPERACIONES*. Novena. Ciudad de México: Universidad Iberoamericana, 2016.
- [3] Wheeler Mykel J. Kochenderfer - Tim A. *Algorithms for Optimization*. 1.^a ed. The MIT Press, 2019.
- [4] Tsitsiklis Dimitris Bertsimas - John N. Tsitsiklis - John. *Introduction to Linear Optimization*. 1.^a ed. Athena Scientific, 1997.
- [5] Muñoz Luis A. Moncayo-Martínez - David Fernando. “Un Sistema de Apoyo para la Enseñanza del Método Simplex y su Implementación en Computadora”. En: *Versión On-line* 11.6 (2018).
- [6] “User’s Manual for CPLEX”. En: (2021). URL: <https://www.ibm.com/docs/en/icos/12.8.0.0?topic=cplex-users-manual>.
- [7] Kernighan Robert Fourer - David M. Gay -Brian W. *AMPL: A Modeling Language for Mathematical Programming*. 2.^a ed. Cengage Learning, 2002.
- [8] Christian Kirches Hans Georg Bock Sven Leyffer. “Modeling Mixed-Integer Constrained Optimal Control Problems in AMPL”. En: *IFAC Proceedings Volumes*. Vol. 45. 2012, págs. 1124-1129.
- [9] Irazábal Ricardo Rosenfeld - Jerónimo. *Computabilidad, complejidad computacional y verificación de programas*. Segunda. Argentina: EDULP, 2013.
- [10] Mañas José A. *Análisis de Algoritmos – Complejidad*. Dept. de Ingeniería de Sistemas Telemáticos, 2017.

- [11] Hernández Adrián. “Solución al problema P vs NP?” En: (2017). URL: <https://mlearninglab.com/2017/08/17/solucion-al-problema-p-vs-np/>.
- [12] Camagüey Universidad. “Application of Operations Research in the Distribution of COVID-19-Related Resources”. En: (2020). URL: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2306-91552020000200086.
- [13] Estado de Hidalgo Universidad Autónoma del. “Enfoque para la solución de un problema de transporte en la cadena de suministro agroalimentaria de la cebada en México”. En: (2017). URL: <https://www.uaeh.edu.mx/scige/boletin/sahagun/n7/a2.html>.
- [14] Noraini Abdullah Ting kien Hua. “Operation Management on transportation and distribution problem”. En: (2017). URL: https://www.researchgate.net/publication/314086499_OPERATION_MANAGEMENT_ON_TRANSPORTATION_AND_DISTRIBUTION_PROBLEM.
- [15] Karel Jeřábek Mária Stopková. “Using the Operations Research Methods to Address Distribution Tasks at a City Logistics Scale”. En: *Horizons of Autonomous Mobility in Europes*. Vol. 44. 2020, págs. 348-355.
- [16] Lumír Pečený Pavol Meško Rudolf Kampf. “Optimisation in Transport and Logistic Processes”. En: *Transportation Research Procedia*. Vol. 44. 2020, págs. 15-22.
- [17] Science Direct. *Nearest Neighbor Method*. <https://www.sciencedirect.com/topics/mathematics/nearest-neighbor-method>. 2012.
- [18] “Vogel’s Approximation Method (VAM)”. En: *Encyclopedia of Operations Research and Management Science*. Ed. por Gass Saul I. y Fu Michael C. Boston, MA: Springer US, 2013, págs. 1630-1630. ISBN: 978-1-4419-1153-7. DOI: [10.1007/978-1-4419-1153-7_200907](https://doi.org/10.1007/978-1-4419-1153-7_200907). URL: https://doi.org/10.1007/978-1-4419-1153-7_200907.
- [19] “Ordinary differential equation (ODE)”. En: *A First Course in Differential Equations with Modeling Application*. Ed. por Zill Dennis G. Cengage Learnin, 2012. ISBN: 978-1-285-40110-2. DOI: [10.1007/978-1-4419-1153-7_200907](https://doi.org/10.1007/978-1-4419-1153-7_200907).
- [20] MathWorks. *Differential algebraic equation (DAE)*. <https://www.mathworks.com/help/matlab/math/solve-differential-algebraic-equations-daes.html>.

- [21] Schweiger C.A. y Floudas C.A. “Process Synthesis, Design, and Control: A Mixed-Integer Optimal Control Framework”. En: *IFAC Proceedings Volumes* 31.11 (1998), págs. 191-196.
- [22] Nader Al Theeb Hazem J.Smadi Tarek H.Al-Hawari. “Optimization of vehicle routing with inventory allocation problems in Cold Supply Chain Logistics”. En: *Computers Industrial Engineering*. Vol. 142. 2020.
- [23] Govind Shay, Sharma Govind y Kumar Amitesh. “Solving a travelling salesman problem with heuristic model approach and comparing with AMPL solution Solving a travelling salesman problem with heuristic model approach and comparing with AMPL solution”. En: *Iraqi Journal of Science* 17 (abr. de 2021), pp]1-8[.
- [24] Joshi A.S. y col. “Cuckoo Search Optimization- A Review”. En: *Materials Today: Proceedings* 4.8 (2017), págs. 7262-7269.
- [25] Saghand Payman Ghasemi y Charkhgard Hadi. “Exact solution approaches for integer linear generalized maximum multiplicative programs through the lens of multi-objective optimization”. En: *Computers Operations Research* 137 (2022), pág. 105549. ISSN: 0305-0548. DOI: <https://doi.org/10.1016/j.cor.2021.105549>.
- [26] Renata A.Konrad Andrew C.Trapp Timothy M.Palmbach. “Overcoming human trafficking via operations research and analytics: Opportunities for methods, models, and applications”. En: *European Journal of Operational Research*. Vol. 259. 2017, págs. 733-745.
- [27] Muzamil Mohd y col. *Application of operation research techniques in agricultural engineering*. Sep. de 2022.
- [28] Baker Barrie M. y Ayechew M.A. “A genetic algorithm for the vehicle routing problem”. En: *Computers Operations Research* 30.5 (2003), págs. 787-800.
- [29] MathWorks. *What Is the Genetic Algorithm?* <https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>.
- [30] Science Direct. *Tabu Search*. <https://www.sciencedirect.com/topics/computer-science/tabu-search>.
- [31] Bertsimas Dimitris y Tsitsiklis John. “Simulated Annealing”. En: *Statistical Science* 8.1 (1993), págs. 10-15.

- [32] Pierre Hansen Nenad Mladenović y Pérez José A. Moreno. “Variable neighbourhood search: methods and applications”. En: *Annals of Operations Research* 175 (2010), págs. 367-407.
- [33] “Crane Scheduling of Steel-making and Continuous Casting Process Using the Mixed-Timed Petri Net Modelling Via CPLEX Optimization”. En: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, págs. 9482-9487. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20110828-6-IT-1002.00170>.
- [34] Manupati Vijaya Kumar y col. “Convalescent plasma bank facility location-allocation problem for COVID-19”. En: *Transportation Research Part E: Logistics and Transportation Review* 156 (2021), pág. 102517. ISSN: 1366-5545. DOI: <https://doi.org/10.1016/j.tre.2021.102517>. URL: <https://www.sciencedirect.com/science/article/pii/S1366554521002787>.
- [35] Fatahayu Nafi y col. “Optimization of Blood Bag Distribution Routes Using AMPL Software and Nearest Neighbor Algorithm (Case Study of the Indonesian Red Cross Jakarta)”. En: *Jurnal Logistik Indonesia* 6 (abr. de 2022), págs. 1-12.

ANEXOS

Anexo 1

Código en AMPL del modelo 4

```
param I; #Productores
param J; #Plantas
param K; #Consumidores

param G; #Tipos de Grano
param T; #Periodos de Tiempo
param C; #Tipos de Camiones

param P; #Periodos de Demanda/Cozecha

param CostTransp;
param CostEspera;

#####
set Productores = 1..I;
set Plantas= 1..J;
set Consumidores= 1..K;
set Granos= 1..G;
set Tiempos= 1..T;
set Camiones= 1..C;
set CantPeriodos= 1..P;
#####
param FinPeriodos{p in CantPeriodos};
```

```

param InicioPeriodos{p in CantPeriodos};

param DistPrPl{i in Productores, j in Plantas};
param DistPlCo{j in Plantas, k in Consumidores};

param Oferta {i in Productores,
g in Granos, p in CantPeriodos};
param Demanda{k in Consumidores, g in Granos, p in CantPeriodos};

param TiempoProduccion{j in Plantas};
param AccesoAPlanta{j in Plantas, c in Camiones};
param CilosEnPlanta{j in Plantas, g in Granos};

param Descarga{c in Camiones};
param VelMed;

var x {i in Productores, j in Plantas, g in Granos,
t in Tiempos, c in Camiones} integer >= 0;
var y {j in Plantas, k in Consumidores, g in Granos,
t in Tiempos, c in Camiones} integer >= 0;
var E {j in Plantas, t in Tiempos};

minimize costo:
sum{j in Plantas} sum{i in Productores} sum{g in Granos}
sum{t in Tiempos} sum{c in Camiones}
(x[i,j,g,t,c]*DistPrPl[i,j]*CostTransp) +
sum{j in Plantas} sum{k in Consumidores} sum{g in Granos}
sum{t in Tiempos} sum{c in Camiones}
(y[j,k,g,t,c]*DistPlCo[j,k]*CostTransp) +
sum{j in Plantas} sum{t in Tiempos}
(E[j,t]*(E[j,t]+1)*CostEspera)/2
;

subject to DemandaCumplida {k in Consumidores, g in Granos,
p in CantPeriodos}:
sum{j in Plantas} sum{c in Camiones} sum{t in 1..FinPeriodos[p]}

```

```

    if (t+(DistPlCo[j,k]/VelMed)) <= FinPeriodos[p] then
        y[j,k,g,t,c] = sum{u in 1..p} Demanda[k,g,u]
;

subject to CumplirConPlantas {j in Plantas, g in Granos,
t in Tiempos}:
    sum{c in Camiones} sum{i in Productores}
    if ((round((t-DistPrPl[i,j]/VelMed)-TiempoProduccion[j])) > 0)
    then
        x[i,j,g,round(t-(DistPrPl[i,j]/VelMed)-TiempoProduccion[j]),c]
        >= sum{c in Camiones} sum{k in Consumidores} y[j,k,g,t,c]
;

subject to OfertaCumplida {i in Productores, g in Granos,
p in CantPeriodos}:
    sum{j in Plantas} sum{c in Camiones}
    sum{t in InicioPeriodos[p]..FinPeriodos[p]}
        x[i,j,g,t,c] <= Oferta[i,g,p]
;

subject to TiposDeEspera{j in Plantas, t in Tiempos}:
    E[j,t] = sum{i in Productores} sum{g in Granos}
        sum{c in Camiones}
    if (round(t-(DistPrPl[i,j]/VelMed)) > 0 then
        x[i,j,g,round((t-(DistPrPl[i,j]/VelMed)),c)*Descarga[c]
    else
        0
;

subject to AccesibilidadAPlanta{ i in Productores, j in Plantas }:
    sum{g in Granos} sum{t in Tiempos} sum{c in Camiones}
        x[i,j,g,t,c] * AccesoAPlanta[j,c] = 0
;

```

```

subject to CapacidadEnSilos{ t in Tiempos, g in Granos,
j in Plantas }:
    sum{i in Productores} sum{k in Consumidores} sum{h in 1..t}
    sum{c in Camiones}
        (x[i,j,g,h,c] - y[j,k,g,h,c]) <= CilosEnPlanta[j,g]
;

```

Anexo 2

Código en Node Js para la generación de pruebas

2.1. Código del generador de datos

```
const { checkPrimeSync } = require('crypto');

var rutaTemplate = 'Template.txt'
const fs = require('fs')
Folders = [ "Cilos", "Demanda", "Ofertas", "Dist Planta Consumidor",
"Productores"]
k=15;
///
```

```
MinDistPlCo = 1;  
MinOfertas = 10;  
MinDemandas = 3;  
MinCilo = 500;
```

```
MaxPr = 1000;  
MaxPl = 15;  
MaxCo = 6;  
MaxGr = 400;  
MaxCa = 8;  
MaxTi = 1000;  
MaxZa = 10;
```

```
MaxTp = 10;  
MaxCosTr = 100;  
MaxCosEs = 20;  
MaxDistPrPl = 100;  
MaxDistPlCo = 100;  
MaxOfertas = 150000;  
MaxDemandas = 150000;  
MaxCilo = 3500;
```

```
function GenTestFile(FileName, UsarVariantes, MinPrVar,  
MinPlVar, MinCoVar, MinGrVar, MinCaVar, MinTiVar,  
MinZaVar, MinTpVar, MinCosTrVar, MinCosEsVar,  
MinDistPrPlVar, MinDistPlCoVar, MinOfertasVar,  
MinDemandasVar, MinCiloVar){  
    ///Rangos  
    if (!UsarVariantes){  
        MinPr = MinPrVar  
        MinPl = MinPlVar  
        MinCo = MinCoVar  
        MinGr = MinGrVar  
        MinCa = MinCaVar  
        MinTi = MinTiVar  
        MinZa = MinZaVar;
```

```

MinTp = MinTpVar;
MinCosTr = MinCosTrVar;
MinCosEs = MinCosEsVar;
MinDistPrPl = MinDistPrPlVar;
MinDistPlCo = MinDistPlCoVar;
MinOfertas = MinOfertasVar;
MinDemandas = MinDemandasVar;
MinCilo = MinCiloVar;
}

///

```

```

data = data.replace ('[MaxGranos]', MaxGranos)
data = data.replace ('[MaxCamiones]', MaxCamiones)
data = data.replace ('[MaxTiempos]', MaxTiempos)
data = data.replace ('[MaxZafras]', MaxZafras)

data = data.replace ('[CostTransp]', CostTransp)
data = data.replace ('[CostEspera]', CostEspera)

//Periodos
var InicioPeriodos = '';
var FinPeriodos = '';
var Delta= Math.floor (MaxTiempos/MaxZafras) //5
for (var i=1; i <MaxZafras+1; i++){
    if (i==1){
        InicioPeriodos+= '\n' + i + ' ' + '1';
    }else{
        InicioPeriodos+= '\n' + i + ' ' + Delta*(i-1);
    }
    if (i==MaxZafras){
        FinPeriodos+= '\n' + i + ' ' + MaxTiempos;
    }else{
        fin= (Delta*i)-1
        FinPeriodos+= '\n' + i + ' ' + fin;
    }
}
data = data.replace ('[InicioPeriodos]', InicioPeriodos)
data = data.replace ('[FinPeriodos]', FinPeriodos)

//Tiempos de Produccion
var TiempoProduccion = '';
for (var i=1; i <MaxPlantas+1; i++){
    TiempoProduccion+= '\n' + i + ' ' +
    getRandomInt (MinTp, MaxTp, UsarVariantes);
}
data = data.replace ('[TiempoProduccion]', TiempoProduccion)

```

```

//Distancias PrPl
var PlantasList = '';
for(var i=1; i <MaxPlantas+1; i++){
    PlantasList+= ' ' + i;
}
data = data.replace('[PlantasList]', PlantasList)

var DistPrPl = '';
for(var i=1; i <MaxProductores+1; i++){
    var DistanciaPrPl = ''
    for(var j=1; j <MaxPlantas+1; j++){
        DistanciaPrPl+=' ' + getRandomInt(MinDistPrPl,
            MaxDistPrPl, UsarVariantes);
    }
    DistPrPl+= '\n' + i + ' ' + DistanciaPrPl;
}
data = data.replace('[DistPrPl]', DistPrPl)

//Distancias PlCo
var ConsumidoresList = '';
for(var i=1; i <MaxConsumidores+1; i++){
    ConsumidoresList+= ' ' + i;
}
data = data.replace('[ConsumidoresList]', ConsumidoresList)
var DistPlCo = '';
for(var i=1; i <MaxPlantas+1; i++){
    var DistanciaPlCo = ''
    for(var j=1; j <MaxConsumidores+1; j++){
        DistanciaPlCo+=' ' + getRandomInt(MinDistPlCo,
            MaxDistPlCo, UsarVariantes);
    }
    DistPlCo+= '\n' + i + ' ' + DistanciaPlCo;
}
data = data.replace('[DistPlCo]', DistPlCo)

//Ofertas

```

```

var GranosList = '';
for (var i=1; i <MaxGranos+1; i++){
    GranosList+= ' ' + i;
}
var Oferta = '';
for (var i=1; i <MaxZafras+1; i++){
    Oferta+='[*,*,'+i+']:'+ GranosList +':='
    for (var j=1; j <MaxProductores+1; j++){
        var ListaOfertas = ''
        for (var z=1; z <MaxGranos+1; z++){
            ListaOfertas+=' ' + getRandomInt (MinOfertas ,
            MaxOfertas , UsarVariantes)
        }
        Oferta+= '\n'+ j+ ListaOfertas;
    }
    Oferta+= '\n'
}
data = data.replace ( '[Oferta]' , Oferta)

//Demandas
var GranosList = '';
for (var i=1; i <MaxGranos+1; i++){
    GranosList+= ' ' + i;
}
var Demanda = '';
for (var i=1; i <MaxZafras+1; i++){
    Demanda+='[*,*,'+i+']:'+ GranosList +':='
    for (var j=1; j <MaxConsumidores+1; j++){
        var ListaDemandas = ''
        for (var z=1; z <MaxGranos+1; z++){
            ListaDemandas+=' ' + getRandomInt (MinDemandas ,
            MaxDemandas , UsarVariantes)
        }
        Demanda+= '\n'+ j+ ListaDemandas;
    }
}

```

```

        Demanda+= '\n'
    }
    data = data.replace ('[Demanda]', Demanda)

//Velocidad de Descarga
var Descarga= '';
for(var i=1; i <MaxCamiones+1; i++){
    Descarga+=' \n' + i + ' ' +
        getRandomInt(1, 100, UsarVariantes);
}
data = data.replace ('[Descarga]', Descarga)

//AccesoAPlanta
var CamionesList = '';
for(var i=1; i <MaxCamiones+1; i++){
    CamionesList+= ' ' + i;
}
data = data.replace ('[CamionesList]', CamionesList)
var AccesoAPlanta = '';
for(var i=1; i <MaxPlantas+1; i++){
    var TieneAcceso = ''
    for(var j=1; j <MaxCamiones+1; j++){
        TieneAcceso+= ' ' + getRandomInt(0, 1, UsarVariantes);
    }
    AccesoAPlanta+= '\n' + i + ' ' + TieneAcceso;
}
data = data.replace ('[AccesoAPlanta]', AccesoAPlanta)

//Silos
var GranosList = '';
for(var i=1; i <MaxGranos+1; i++){
    GranosList+= ' ' + i;
}
data = data.replace ('[GranosList]', GranosList)
var CilosEnPlanta = '';

```

```

for (var i=1; i <MaxPlantas+1; i++){
    var VolumenEnCilo = ''
    for (var j=1; j <MaxGranos+1; j++){
        if (TieneCilo(50)){// baterias de 10/16 silos
            VolumenEnCilo+=' ' + getRandomInt(MinCilo, MaxCilo,
            UsarVariantes);
        }else{
            VolumenEnCilo+=' ' + 0;
        }
    }
    CilosEnPlanta+= '\n' + i + ' ' + VolumenEnCilo;
}
data = data.replace('[CilosEnPlanta]', CilosEnPlanta)

if (!fs.existsSync('../TestyRunners/')){
    fs.mkdirSync('../TestyRunners/');
}
fs.writeFile('../TestyRunners/'+FileName+'.dat', data, function (
    if (err) return console.log(err);
    console.log(FileName+' generado;');
});

})
}

function getRandomInt(min, max, befair) {
    if (!befair)
        return min;
    return Math.floor(Math.random() * (max - min)) + min;
}

```

```

function TieneCilo(Probabilidad){
    var temp = Math.random() * 100;
    if (temp < Probabilidad)
        return true;
    else
        return false;
}

//GenTestFile('Test'+”Aux”,false , 10, 5);
function resetVars(){
    SmallNum=1;
    BigNum=20;
    ConstNum= 1;

    MinPrVar      = SmallNum;      //Productores
    MinPlVar      = SmallNum;      //Plantas
    MinCoVar      = SmallNum;      //Consumidores
    MinGrVar      = SmallNum;      //Granos
    MinCaVar      = SmallNum;      //Camiones

    MinTiVar      = BigNum;        //Tiempo
    MinZaVar      = SmallNum;      //Zafras
    MinTpVar      = ConstNum;

//Tiempo de Produccion

    MinCosTrVar   = ConstNum;
//Costo de Transporte
    MinCosEsVar   = ConstNum;     //Costo de Espera

    MinDistPrPlVar = ConstNum;
//Distancia entre Productores y Plantas
    MinDistPlCoVar = ConstNum;
//Distancia entre Plantas y Consumidores

    MinOfertasVar = ConstNum;     //Ofertas
    MinDemandasVar = ConstNum;    //Demandas

```

```

    MinCiloVar      = BigNum;          // Cilos
    k—
}

function delay(time) {
    return new Promise(resolve => setTimeout(resolve, time));
}

run();

async function run() {
    resetVars();
    for (var i=1; i<=1; i++){
        GenTestFile('Test', true,
100, 5, 100, 1, 5, 100, 1, 1, 1, 1, 1, 1, 10, 10, 200);
        await delay(1000);
    }
}

```

2.2. T mplate de datos utilizado por el generador

```

data;

param I:=[MaxProductores]; #Cantidad de Productores
param J:=[MaxPlantas]; #Cantidad de Plantas
param K:=[MaxConsumidores]; #Cantidad de Consumidores

param G:=[MaxGranos]; #Cantidad de Tipos de Granos
param T:=[MaxTiempos]; #Cantidad de Periodos de Tiempos
param C:=[MaxCamiones]; #Cantidad de Tipos de Camiones

param P:=[MaxZafras]; #Cantidad de Zafras

```

```

#Momentos en el tiempo donde incia una Zafra
param InicioPeriodos:= [InicioPeriodos];

#Momentos en el tiempo donde termina una Zafra
param FinPeriodos:= [FinPeriodos];

#Cantidad de tiempo que cada Planta requiere para trabajar
#con los granos
param TiempoProduccion:=[TiempoProduccion];

#Cosato de Transporte Promedio
param CostTransp:=[CostTransp];

#Cosato de Espera Promedio
param CostEspera:=[CostEspera];

#Distancia entre Productores [X,*] y Plantas [*,Y]
param DistPrPl:[PlantasList] := [DistPrPl];

#Distancia entre Plantas [X,*] y Consumidores [*,Y]
param DistPlCo: [ConsumidoresList] := [DistPlCo];

#Cantidad de Granos [*,Y,*] que ofrece el Productor [X,*,*] en la
#Zafra [*,*,Z]
param Oferta [Oferta];

#Cantidad de Granos [*,Y,*] que demanda el Consumidor [X,*,*] en la
#Zafra [*,*,Z]
param Demanda:=[Demanda];

#Velocidad de Descarga de los tipos de Camiones
param Descarga:= [Descarga];

#Si la Planta [X,*] acepta la entrada de Camiones [*,Y] = 0 y 1

```

```
#en caso contrario
param AccesoAPlanta: [CamionesList] := [AccesoAPlanta];

#Cantidad de Granos [*,Y] que puede acumular la Planta [X,*]
param CilosEnPlanta: [GranosList] :=[CilosEnPlanta];

#La Velocidad Promedio de los Camiones
param VelMed:= 80;

end;
```

Anexo 3

Datos de pruebas y resultados

Se listan en las siguientes referencias los diferentes archivos de modelos y pruebas realizadas con sus respectivos resultados.

- [Modelos en AMPL](#)
- [Pruebas de Validación](#)
- [Pruebas de Relajaciones](#)
- [Pruebas de Sensibilidad](#)