



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis del problema de dimensionado de lotes con remanufactura e incertidumbre en la demanda y los retornos

Fernando Islas

Maestría en Investigación de Operaciones
Depto. de Investigación Operativa, Instituto de Computación
Facultad de Ingeniería, Universidad de la República, Uruguay

Montevideo Uruguay

Setiembre de 2021



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis del problema de dimensionado de lotes con remanufactura e incertidumbre en la demanda y los retornos

Fernando Islas

Tesis de Maestría presentada a la Maestría en Investigación de Operaciones, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magister en Investigación de Operaciones.

Directores: Pedro Piñeyro

Carlos Testuri

Director Académico: Pedro Piñeyro

Montevideo Uruguay

Junio de 2021

Islas De Maio, Fernando Francisco

Análisis del problema de dimensionado de lotes con remanufactura e incertidumbre en la demanda y los retornos / Fernando Francisco Islas De Maio. - Montevideo: Universidad de la República, Facultad de Ingeniería, 2021.

VIII, 134 p. 29,7cm.

Directores:

Pedro Piñeyro

Carlos Testuri

Director académico:

Pedro Piñeyro

Tesis de Maestría - Universidad de la República, Facultad de Ingeniería, Maestría en Investigación de Operaciones.

Referencias bibliográficas: p. 127 - 134.

1. Problema de Dimensionamiento de Lotes, 2. Remanufactura, 3. Programación Estocástica, 4. Reformulaciones, 5. Inecuaciones Válidas.

I. Piñeyro, Pedro; Testuri, Carlos. II. Universidad de la República, Maestría en Investigación de Operaciones. III. Análisis del problema de dimensionado de lotes con remanufactura e incertidumbre en la demanda y los retornos.

INTEGRANTES DEL TRIBUNAL DE LA DEFENSA DE TESIS

Prof. José Framiñán (Universidad de Sevilla)

Prof. Soledad Gutiérrez (UDELAR)

Prof. Antonio Mauttone (UDELAR)

Agradecimientos

Agradezco sinceramente a mis directores de tesis, Dr. Pedro Piñeyro y Dr. Carlos Testuri, por su apoyo constante y asesoría siempre dispuesta durante todo el desarrollo de la maestría. Quiero agradecer también a los distintos profesores del Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República que han tenido un rol importante en mi formación. Un reconocimiento especial a mi familia, que me ha apoyado en los momentos adversos. Finalmente, deseo agradecer a la Agencia Nacional de Investigación e Innovación (ANII) por el apoyo brindado a través de la beca para Posgrados Nacionales.

Resumen

El problema económico de dimensionamiento de lotes con remanufactura (ELSR por sus siglas en inglés) es una extensión del problema tradicional, en el que la demanda también puede satisfacerse mediante la remanufactura de productos usados (también llamados retornos) que son devueltos al origen. En la práctica, las decisiones de planificación deben tomarse en condiciones de incertidumbre. En este contexto, las decisiones deben tener en cuenta todas las realizaciones posibles de los parámetros de interés. Nuestro trabajo considera dos variantes del ELSR con incertidumbre en las cantidades de demanda y de retornos: con una sola línea tanto para la producción como para la remanufacturación, y con líneas dedicadas para cada una de estas dos actividades. Como parte de la revisión de la literatura, se presentan diferentes reformulaciones existentes y enfoques de solución para el caso determinista y de variantes estocásticas. Esta revisión se amplía mediante el diseño de formulaciones de programación entera mixta estocástica para las que se utilizan árboles de escenarios. La formulación natural del problema se reformula como un problema de localización de instalaciones, el cual se ha demostrado que es capaz de brindar buenos resultados para ELSR en su versión determinista. Nuestro objetivo luego, fue fortalecer las formulaciones naturales mediante la incorporación de inecuaciones válidas de camino y de árbol. También presentamos una modificación de la heurística relax-and-fix para el problema. Finalmente, presentamos los resultados obtenidos de la experimentación numérica llevada a cabo para el caso de líneas dedicadas, junto con las conclusiones y algunas posibles líneas para investigaciones futuras. Los hallazgos más relevantes que surgen del estudio computacional son los siguientes: la formulación con inecuaciones válidas de árbol tiene el potencial de mejorar la relajación a Programación Lineal del problema; en comparación con una formulación natural, el rendimiento de las inecuaciones de camino y de las reformulaciones de localización de las instalaciones disminuyen a medida que aumenta el horizonte de planificación del problema; el procedimiento heurístico sugerido parece ser prometedor para problemas de gran tamaño.

Palabras clave: Problema de Dimensionamiento de Lotes; Remanufactura; Programación Estocástica; Reformulaciones; Inecuaciones Válidas.

Abstract

The Economic Lot Sizing Problem with Remanufacturing (ELSR) is an extension of the economic lot-sizing problem, in which demand can also be satisfied by remanufacturing used products returned to the origin. In practice, planning decisions may be taken under uncertainty. In this context, decisions should take into account all possible realizations of the parameters. Our work addresses two stochastic variants of this problem: with a single production line, and with dedicated production lines. As part of a literature review, reformulations and solution approaches for the deterministic case and stochastic variants are presented. This review is extended by designing stochastic mixed integer programming formulations for the variants using scenario trees. The natural formulation of the problem is reformulated as a facility location problem, which has previously been proven to provide good results for ELSR. We aim to strengthen the natural formulations by incorporating path and tree valid inequalities. We also present a modification of the relax-and-fix heuristic. Finally, we present a numerical study for the case of dedicated production lines, along with the conclusions and offer some directions for future research. The key findings that arise from the computational study are: the formulation with tree valid inequalities has the potential to improve the LP relaxation of the problem; performance of path inequalities and facility location reformulations decrease with a high number of periods, as compared to a natural formulation; the heuristic procedure seems to be promising for higher problem sizes.

Keywords: Lot Sizing Problem; Remanufacturing; Stochastic Programming; Reformulations; Valid Inequalities.

Índice general

Índice de tablas	x
1. Introducción	1
2. Conceptos y Revisión de la Literatura	5
2.1. Problema de dimensionado de lotes	7
2.1.1. Formulación del problema de dimensionado de lotes dinámico	7
2.1.2. Resolución del problema de dimensionado de lotes dinámico	8
2.1.3. Costos de Wagner-Within	9
2.1.4. Variantes	9
2.1.5. Clasificación de problemas de dimensionado de lotes	12
2.2. Análisis del dimensionado de lotes con remanufactura determinista	14
2.2.1. Variantes que incorporan remanufactura de retornos	14
2.2.2. Primeros modelos del problema de dimensionado de lotes con remanufactura	15
2.2.3. Formulación determinista de ELSRs y ELSRj	17
2.2.3.1. Definición de conjuntos, parámetros y variables de ELSRs y ELSRj	17
2.2.3.2. Formulación ELSRs	18
2.2.3.3. Formulación ELSRj	19
2.2.4. Complejidad de ELSRj y ELSRs	20
2.3. Estrategias de resolución para el caso determinista	22
2.3.1. Ramificado y acotamiento	23
2.3.2. Formulaciones extendidas	24
2.3.2.1. Camino mínimo	25
2.3.2.2. Localización de instalaciones	26
2.3.3. Planos de corte	27
2.3.3.1. Inecuaciones válidas para el DLSP	28

2.3.3.2.	Inecuaciones válidas para el problema con remanufactura determinista	29
2.3.4.	Equivalencias entre formulaciones	33
2.3.5.	Heurísticas	33
2.3.5.1.	Método relax-and-fix	34
2.4.	Análisis de variantes del dimensionado de lotes con remanufactura estocástico	36
2.4.1.	Problemas estocásticos generales	37
2.4.1.1.	Primeros modelos	37
2.4.1.2.	Formulación genérica basada en árboles de escenarios	38
2.4.2.	Dimensionado de lotes estocástico	39
2.4.3.	Literatura de dimensionado de lotes estocástico para problemas con remanufactura	43
3.	Dimensionado de Lotes con Remanufactura Estocástico	45
3.1.	Descripción del problema	47
3.2.	Representación mediante árboles de escenarios	49
3.3.	Formulación natural	51
3.3.1.	Modelo con configuración en líneas de producción separadas (costos de setup separados)	53
3.3.2.	Modelo con configuración en una línea de producción común (costo de setup común)	55
3.3.3.	Complejidad	56
4.	Reformulaciones e Inecuaciones Válidas	57
4.1.	Reformulación localización de instalaciones	58
4.1.1.	Localización de instalaciones setup separado	59
4.1.2.	Localización de instalaciones setup común	61
4.2.	Reformulación mediante el agregado de inecuaciones válidas	63
4.2.1.	Inecuaciones válidas para SELSRs	64
4.2.1.1.	Inecuaciones equivalentes sin variables de inventario para SELSRs	66
4.2.2.	Inecuaciones válidas para SELSRj	68
4.2.2.1.	Inecuaciones sin variables de inventario SELSRj	68
4.3.	Restricciones de árbol	69
4.3.1.	Emparejamiento	69
4.3.2.	Restricciones de árbol para problemas estocásticos generales	70
4.3.3.	Restricciones de árbol para SELSRs	72

5. Resolución Heurística	79
5.1. Descripción del método relax-and-fix	79
5.2. Esquema relax-and-fix tradicional	82
5.3. Descomposición en subproblemas dentro de un período	83
5.4. Implementación relax-and-fix con ventana central variable	85
6. Estudio Computacional	89
6.1. Generación de casos de prueba	89
6.2. Selección de datos y configuración de ejecución	91
6.3. Estudio para instancias de tamaño pequeño	94
6.3.1. Interpretación de resultados	95
6.3.2. Análisis para seis períodos	99
6.3.3. Análisis para ocho períodos	99
6.4. Estudio para instancias de tamaño mediano	101
6.5. Estudio para instancias de tamaño grande	107
6.5.1. Desempeño de las reformulaciones para instancias de tamaño grande	107
6.5.2. Desempeño de la heurística para instancias de tamaño grande . . .	113
7. Conclusiones	117
7.1. Reformulación con inecuaciones válidas de camino	117
7.2. Reformulación con inecuaciones válidas de árbol	118
7.3. Heurística	119
7.4. Dificultad de resolución de las instancias	120
7.5. Comparación de desempeño de estrategias de resolución	121
8. Trabajos Futuros	124
Bibliografía	127

Índice de tablas

3.1. Notación de parámetros y variables para los problemas SELSRs y SELSRj .	52
5.1. Ejemplos de ejecución del pseudocódigo en los que se muestra, para tres combinaciones de valores de γ y δ , el primer y último período de la ventana central de la heurística en cada iteración.	87
6.1. Pruebas con 6 períodos y costo de configuración 125	95
6.2. Pruebas con 6 períodos y costo de configuración 250	96
6.3. Pruebas con 6 períodos y costo de configuración 500	96
6.4. Pruebas con 8 períodos y costo de configuración 125	97
6.5. Pruebas con 8 períodos y costo de configuración 250	97
6.6. Pruebas con 8 períodos y costo de configuración 500	98
6.7. Pruebas con 10 períodos y costo de configuración 125	103
6.8. Pruebas con 10 períodos y costo de configuración 250	104
6.9. Pruebas con 10 períodos y costo de configuración 500	105
6.10. Porcentaje de soluciones factibles para 12 períodos y costo de configuración 250	108
6.11. Porcentaje de soluciones factibles para 12 períodos y costo de configuración 500	109
6.12. Pruebas con 12 períodos y costo de configuración 125	110
6.13. Pruebas con 12 períodos y costo de configuración 250	111
6.14. Pruebas con 12 períodos y costo de configuración 500	112
6.15. Comparación desempeño de la heurística con formulación natural, con 12 períodos y costo de configuración 125	115
6.16. Comparación desempeño de la heurística con formulación natural, con 12 períodos y costo de configuración 250	115
6.17. Comparación desempeño de la heurística con formulación natural, con 12 períodos y costo de configuración 500	116

Capítulo 1

Introducción

La planificación de la producción consiste en la organización de recursos y actividades para transformar insumos en productos finales para atender una demanda de forma eficiente. Un problema ampliamente estudiado dentro de esta área es el Problema de Dimensionado de Lote Dinámico (Dynamic Lot Sizing Problem, DLSP). El DLSP consiste en determinar el plan de producción y almacenamiento de mínimo costo de un producto para satisfacer una demanda periódica en un horizonte de tiempo discreto. Esta demanda puede ser atendida tanto con producción del período, o inventario de períodos anteriores. En cada período del horizonte de planificación se puede incurrir en costos asociados a la cantidad de productos producidos, a la activación ó configuración de la producción, y a almacenar productos en el inventario. El interés en los problemas de dimensionado de lotes generalmente se presenta cuando se debe incurrir en un costo fijo, a raíz de la necesidad de configurar, instalar o calibrar maquinaria. Por un lado, se buscará utilizar lotes lo más grande posibles para evitar tener que llevar a cabo muchas configuraciones, y disminuir así la cantidad de veces que se incurre en el costo fijo. Sin embargo, a medida que aumenta el tamaño del lote producido, también aumentarán los costos de almacenamiento. Por lo tanto, normalmente existe un compromiso entre la configuración de producción y el almacenamiento de producción en inventario. Desde la presentación del DLSP, y la posterior aparición del primer algoritmo eficiente para su resolución presentada por Wagner y Whitin (1958), se han estudiado una gran cantidad de variantes de este problema. Algunos ejemplos de variantes son las que consideran una capacidad de producción finita, niveles de producción, backloging, inventario de productos perecederos, entre otras. Una variante que ha tomado particular interés en los últimos años es el problema de Dimensionado de Lotes con Remanufactura. Esta variante responde a la necesidad de disminuir el impacto negativo de las industrias sobre el medio ambiente y disminuir el desperdicio, sin descuidar las ganancias de las empresas. La legislación sobre el medio ambiente y las oportunidades económicas son algunas de las motivaciones de las

empresas para involucrarse con la remanufactura de productos (Teunter et al., 2006). Algunas de las actividades que pueden ser identificadas con la remanufactura son, por ejemplo, la recolección, reparación, desmontaje de piezas y reemplazo de componentes, con el objetivo de obtener productos con el mismo nivel de calidad que los productos nuevos (Suzanne et al., 2020).

El Problema de Dimensionado de Lotes con Remanufactura es una extensión del DLSP que permite que la demanda se pueda satisfacer remanufacturando artículos/productos usados que son devueltos al origen. A estos artículos se los denomina comúnmente retornos. Una vez que llegan al fin de su ciclo de vida, son devueltos por los clientes, y luego el proceso de remanufactura los transforma en productos cuya calidad se considera tan buena como la de artículos nuevos. De esta manera, los productos nuevos y los remanufacturados pueden ser utilizados indistintamente, por lo que se los denomina artículos listos.

La remanufactura de productos está enmarcada en el concepto de economía circular definido en los trabajos de Homrich et al. (2018) y Reike et al. (2018). La remanufactura atiende las nuevas consideraciones sobre el impacto ambiental de la producción, ya que se disminuye la necesidad de consumir recursos naturales y se reduce la contaminación durante el proceso productivo. A su vez, se pueden obtener importantes beneficios económicos, dado que al reparar los componentes recuperables de los productos se ahorra en la inversión en insumos. Diversas industrias utilizan en la actualidad sistemas de remanufactura como por ejemplo las de producción de maquinaria, instrumentos médicos, motores de autos, computadoras, teléfonos celulares, entre otros (Piñeyro y Viera, 2010).

Teunter et al. (2006) consideran dos variantes del Dimensionado de Lotes con Remanufactura. En una de las variantes, las actividades de manufacturación y remanufactura son llevadas a cabo en distintas líneas de producción, mientras que en la otra variante la línea de producción es compartida.

Muchas decisiones de planificación se toman en condiciones de incertidumbre de algunos parámetros del proceso. Para el problema con remanufactura, toman particular importancia la incertidumbre de la demanda y la cantidad de retornos. Esto se debe a que en general resulta difícil prever que cantidad de artículos serán devueltos por los clientes, y a la dificultad de estimar la demanda de artículos listos. En general se asume que los parámetros con incertidumbre siguen un proceso estocástico de tiempo discreto con espacio de probabilidad finito. Su valor se determina luego de conocer la realización de eventos aleatorios en determinados

períodos del horizonte de planificación. De esta manera, se toman decisiones en cada período progresivamente a medida que se ha resuelto la incertidumbre de algunos de los parámetros. Estas decisiones deben además sopesarse con la incertidumbre futura aún no resuelta y las decisiones posteriores. Retel Helmrich et al. (2014) destacan que la suposición de demanda y retornos deterministas puede ser demasiado restrictiva en algunos contextos.

En este trabajo, se consideran dos variantes estocásticas de Problemas de Dimensionado de Lotes con Remanufactura, con líneas de producción separadas para la manufacturación y remanufacturación ó con una única línea de producción compartida, incorporando incertidumbre en los valores de demanda y retornos. Partiendo de un estudio bibliográfico, se presentan las principales reformulaciones y estrategias de resolución para los casos deterministas y para variantes de problemas estocásticos en el Capítulo 2. En el Capítulo 3 esta revisión se extiende diseñando formulaciones de programación entera-mixta (Mixed Integer Programming, MIP) para las variantes estocásticas mencionadas, utilizando árboles de escenarios. De nuestro conocimiento, este es el primer intento de realizar formulaciones MIP para estos problemas. En el Capítulo 4 se presentan diversas reformulaciones, con la intención de mejorar la formulación natural presentada en el capítulo anterior. En primer lugar, se estudia una formulación extendida, la reformulación de localización de instalación no capacitada. Luego, se estudian reformulaciones que incorporan inecuaciones válidas de camino y de árbol. En el Capítulo 5 se investiga una metodología de resolución heurística computacional para la variante con líneas de producción separadas. Se presentan los detalles de la selección y diseño de una heurística relax-and-fix, así como los distintos aspectos de la implementación computacional. En el estudio computacional del Capítulo 6 se compara la eficiencia de las distintas formulaciones sobre la variante estudiada que considera líneas de producción separadas. Para una versión determinista, se compara la incorporación de distintas inecuaciones válidas presentadas por distintos autores. En particular, en ciertas instancias de datos se observa un beneficio al considerar una adaptación de las inecuaciones (l, S, WW) presentadas recientemente por Ali (2017) para el caso determinista, por sobre la utilizada anteriormente por otros autores. Para la variante estocástica de este trabajo, nos interesa analizar el comportamiento de la adaptación a nuestro problema de estrategias que anteriormente habían dado buenos resultados para casos deterministas. Se observa que a partir de determinados tamaños de problema, el desempeño de las mismas se deteriora. En este sentido, buscamos determinar el tamaño, así como otros parámetros del problema, que dificultan la resolución con estos mecanismos. Por otro lado, nos interesa corroborar si el agregado de inecuaciones de árbol mejora las cotas inferiores del problema relajado

a programación lineal. Finalmente, se busca determinar el rendimiento de la heurística de relax-and-fix presentada.

Capítulo 2

Conceptos y Revisión de la Literatura

En este capítulo se presenta un estudio bibliográfico de los Problemas de Dimensionado de Lotes con Remanufactura, partiendo desde los primeros modelos, hasta llegar a versiones que incorporan incertidumbre en alguno de sus parámetros. Se comienza en la Sección 2.1 presentando los modelos más básicos de Dimensionado de Lotes, con especial énfasis en el Problema de Dimensionado de Lotes Dinámico (Dynamic Lot Sizing Problem, DLSP), desarrollado por Wagner y Whitin (1958). Se presenta su formulación y estrategias de resolución. Además, se detallan otras variantes y clasificaciones para los problemas de dimensionado de lotes. En la Sección 2.2 se mencionan los primeros trabajos que incorporan la posibilidad de remanufacturar retornos y posibles agrupaciones para albergar las principales variantes. Luego, se presentan formulaciones para las dos variantes deterministas del caso de estudio del Capítulo 3. Finalmente, se exponen resultados acerca de la complejidad del problema basados en los trabajos de Retel Helmrich et al. (2014) y Teunter et al. (2006).

En la Sección 2.3, se presentan las principales estrategias de resolución utilizadas para el caso determinista. Para una mejor comprensión del capítulo, se presenta también fundamentos básicos de problemas de programación entera-mixta (MIP) necesarios para la comprensión de las estrategias de resolución.

En la Sección 2.4 se estudian variantes estocásticas de problemas de dimensionado de lotes. Se comienza definiendo el concepto de etapa para problemas de programación estocástica. Se presentan los primeros estudios relacionados a problemas estocásticos de diversos tipos, y los principales métodos de resolución. Luego, se presenta literatura sobre problemas relacionados con el dimensionado de lotes estocásticos, clasificándolos en base a la cantidad de etapas del problema. De nuestro conocimiento, no se han desarrollado estudios del Problema de Dimensionado de Lotes con Remanufactura Estocástico en sus variantes ELSRs y ELSRj

utilizando árboles de escenarios. Por lo tanto, se concluye el capítulo presentando estudios para variantes de dimensionado de lotes con remanufactura estocásticos similares.

2.1. Problema de dimensionado de lotes

Una de las primeras aproximaciones a la planificación de producción con gestión de inventarios es el modelo de la Cantidad Económica de Pedido (Economic Order Quantity, EOQ), presentado por Harris (1913). En este modelo se busca obtener un esquema de producción para un único producto, con una escala de tiempo continua, demanda constante y un horizonte de tiempo infinito. Taft (1918) extiende el modelo anterior a una tasa finita de producción, en el modelo del Lote Económico de Producción (Economic Production Quantity, EPQ). Rogers (1958) presenta el Problema de Planificación de Lote Económico (Economic Lot Scheduling Problem), en el que se busca decidir en que momento y que tamaños de lotes producir en el caso de varios productos que compiten por el uso de maquinaria, manteniendo el supuesto de una demanda invariante en el tiempo. El Problema de Planificación de Lote Económico es NP-hard, y su demostración se puede encontrar en el trabajo de Hsu (1983).

Los primeros modelos de dimensionado de lotes en un horizonte de tiempo discreto fueron los desarrollados por Wagner y Whitin (1958) y Manne (1958). Manne (1958) presenta un modelo con capacidades máximas de producción que permite la producción de varios ítems. Wagner y Whitin (1958) presentan el Problema de Dimensionado de Lote Dinámico (Dynamic Lot Sizing Problem, DLSP), también conocido como problema de Dimensionado de Lotes No Capacitado, siendo este el primer modelo que contempla demanda variante en el horizonte temporal.

2.1.1. Formulación del problema de dimensionado de lotes dinámico

El DLSP trata de la planificación de producción de cantidades de un producto en lotes de diferente tamaño, en un horizonte de tiempo T discretizado en períodos. Para cada período $t \in T$ se tiene una demanda d_t a satisfacer y se incurre en costos unitarios de producción p_t , costos unitarios de almacenar producción en inventario h_t y costos de configurar o activar producción K_t . Se debe decidir el tamaño de lote a producir en cada período x_t y la cantidad de producción a almacenar en inventario al final de cada período I_t de modo de minimizar los costos. Se denota M_t a la demanda restante acumulada de un período t , definida como la suma de las demandas desde t hasta el último de los períodos t_f , de manera que $M_t = \sum_{i=t}^{t_f} d_i$. El modelo MIP resultante es el siguiente:

$$(\text{DLSP}) : \min \sum_{t \in T} (K_t y_t + h_t I_t + p_t x_t) \quad (2.1)$$

s.a :

$$I_{t-1} + x_t = I_t + d_t, \quad t \in T \quad (2.2)$$

$$x_t \leq M_t y_t, \quad t \in T \quad (2.3)$$

$$I_0 = 0, \quad (2.4)$$

$$x_t, I_t \geq 0, y_t \in \{0, 1\}, \quad t \in T \quad (2.5)$$

El objetivo (2.1) es minimizar los costos totales de producción: costos unitarios, costos de almacenar en inventario y costos de configurar producción. Las restricciones (2.2) indican que la demanda de un período se puede satisfacer con producción de ese período, o producción de períodos anteriores, mientras que el excedente de producción se transfiere al período siguiente como inventario. Las restricciones (2.3) son de activación de producción. Para producir en un período, es necesario incurrir en costos de configuración. Para obtener una cota para dichas restricciones, se observa que la producción en un período se ve limitada por la demanda restante acumulada. Las restricciones (2.4) indican que el inventario inicial es nulo. Finalmente se tienen las restricciones de dominio en (2.5). Tanto la cantidad de producción como de inventario al final de cada período son no negativas, y la activación de producción es binaria.

El Problema de Dimensionado de Lotes Dinámico se puede resolver de forma eficiente. En la siguiente Sección se mencionan los estudios más trascendentes de la literatura para la resolución del problema.

2.1.2. Resolución del problema de dimensionado de lotes dinámico

Para la resolución del Problema de Dimensionado de Lotes Dinámico, Wagner y Whitin (1958) desarrollaron un algoritmo de orden polinomial $O(|T|^2)$ basado en programación dinámica. Desde entonces, se han desarrollado extensiones del algoritmo para variantes y para casos particulares del problema. En el trabajo de Zangwill (1968) se presenta una extensión del algoritmo de programación dinámica para costos cóncavos. Federgruen y Tzur (1991), Aggarwal y Park (1993), y Wagelmans et al. (1992) han mejorado la complejidad computacional de los algoritmos, obteniéndose tiempos de resolución $O(|T|)$ y $O(|T|\log|T|)$ para casos particulares.

Barany et al. (1984b) y Barany et al. (1984a) presentan inecuaciones (I, S) y muestran que su inclusión permite describir el casco convexo del conjunto de soluciones del problema. Las inecuaciones son luego incorporadas en un algoritmo de planos de corte. Además, desarrollan una formulación alternativa, como un problema de localización de instalaciones. En la Sección 2.3.3.1 se presenta en mayor detalle las inecuaciones (I, S) , y en la Sección 2.3.3.2 se detallan extensiones para problemas de planificación de producción con remanufactura.

En la siguiente sección se presenta un caso particular del Problema de Dimensionado de Lotes Dinámico, relacionado a los costos de la función objetivo. Este caso ha sido considerado en los estudios de muchos autores, y será de utilidad para analizar la complejidad de distintas extensiones del problema.

2.1.3. Costos de Wagner-Within

En la práctica, las funciones de costos pueden tener una estructura particular, como es el caso de los costos de Wagner-Within:

Definición 1 (Costos de Wagner-Within) *Se dice que un problema de dimensionado de lotes tiene costos de Wagner-Within cuando se cumple: $p_t + h_t \geq p_{t+1} \quad \forall t \in T$*

En un problema con esta estructura, resulta conveniente producir lo más tarde posible. En general, los costos de Wagner-Within permiten desarrollar algoritmos de resolución más eficientes y sencillos. Wagelmans et al. (1992) se refiere a los problemas con costos de Wagner-Within como problemas sin motivos especulativos. Pochet y Wolsey (1994) indican que si se está trabajando con costos de Wagner-Within, esta estructura se puede aprovechar para generar formulaciones extendidas y algoritmos de separación más simples para el DLSP. Los autores demuestran además que el casco convexo tiene $O(|T|^2)$ restricciones, en comparación con las $O(2^{|T|})$ restricciones para el caso general demostradas por Barany et al. (1984a).

2.1.4. Variantes

En esta sección se presentan extensiones o variantes del Problema de Dimensionado de Lotes Dinámico, que adaptan la formulación para poder aplicarla en otras realidades de distintas organizaciones.

Problema multi-item y problema capacitado

Una extensión natural al problema inicial es la de considerar la producción de múltiples ítems. Esta extensión es sencilla a nivel del modelo matemático, incorporándose un índice a las variables y parámetros para representar los ítems. Dado que no existe una restricción que vincule a los distintos ítems, el problema se puede dividir en problemas independientes para cada ítem que se produzca.

En la práctica, las empresas generalmente cuentan con una capacidad limitada de producción. El problema resultante es conocido como el Problema de Dimensionado de Lotes Capacitado (Capacitated Lot Sizing Problem, CLSP). Para casos particulares de este problema en que se produce un único ítem y se tiene una misma capacidad para todos los períodos, el problema se puede resolver con el algoritmo de Florian y Klein (1971) de $O(|T|^4)$. Cuando se produce más de un ítem, modelar capacidades de producción implica incorporar una restricción que vincula los diferentes ítems, de modo que la resolución del problema no es equivalente a dividir el problema en subproblemas independientes. Una estrategia de resolución común para esta clase de problemas consiste en relajar la restricción de capacidad, transformando el mismo en tantos problemas no capacitados independientes como cantidad de ítems se produzcan. Florian et al. (1980) demuestran que contemplar una capacidad limitada convierte a estos problemas de dimensionado en NP-hard, aún cuando se produce un único ítem.

Se han propuesto varias estrategias de resolución para el problema capacitado. Algunos de los métodos exactos explorados han sido la incorporación de inecuaciones válidas a la formulación por Barany et al. (1984b), formulaciones ajustadas basadas en grafos de Eppen y Martin (1987) y estrategias que utilizan el esquema primal-dual de Lozano et al. (1991). Dentro de los métodos aproximados, se han utilizado heurísticas de tipo greedy como por ejemplo las de Dixon y Silver (1981), y las de Maes y Wassenhove (1988). En relación a heurísticas iterativas, se destacan los trabajos de Gopalakrishnan et al. (2001) quienes presentan un algoritmo basado en la metaheurística de Tabú Search para esta clase de problemas y Trigeiro et al. (1989) que desarrollan una heurística basada en la relajación Lagrangeana.

Tiempos de configuración

La variante que incorpora la posibilidad de tener tiempos de configuración puede encontrarse en los trabajos de Maes et al. (1991) y Trigeiro et al. (1989). Dentro de esta extensión del problema se pueden contemplar tiempos de preparación, calibración, limpieza e inspección.

En cada período se tiene una capacidad máxima de tiempo de producción, la cual se ve consumida por los tiempos de producción así como por los tiempos configurar la maquinaria.

Para el CLSP, la factibilidad del problema puede determinarse verificando que en cada período la demanda acumulada no supera a la capacidad acumulada desde ese período hasta el final. Esto permite fácilmente asegurar la existencia o no de alguna solución factible. El problema se vuelve más complejo de resolver cuando son considerados los tiempos de configuración. Trigeiro et al. (1989) indica que para el problema con tiempos de configuración, el problema de factibilidad se vuelve NP-completo. Por lo tanto no sólo no se cuenta con un algoritmo eficiente para hallar una solución óptima, sino que tampoco lo existe para determinar la existencia de una solución factible.

Backlogging

Esta extensión admite la posibilidad de satisfacer la demanda de un período con producción de períodos futuros, es decir, la demanda se satisface con retraso. El primer trabajo que contempla la posibilidad de satisfacer la demanda con atraso es el de Zangwill (1969).

Cuando la producción no se satisface a partir de producción de períodos anteriores o del período actual, generalmente se incurre en una penalidad dependiendo del nivel de atraso.

Inventario de artículos perecederos

En esta variante, los productos almacenados en inventario se pierden a una determinada tasa con el paso del tiempo. Una formulación para este tipo de problemas puede encontrarse en Nahmias (1982). Hsu (2000) propone un algoritmo de programación dinámica $O(|T|^4)$ para el problema, y estudian casos especiales del modelo general que se pueden resolver con un orden de complejidad computacional reducido.

Problema de dimensionado discreto

La variante que incorpora dimensionado discreto fue propuesta por Fleischmann (1990). En esta variante, si se activa la producción en un período debe ser al tope de la capacidad disponible, en lo que se conoce como política all-or-nothing, a diferencia de las variantes tradicionales donde es posible graduar el nivel de producción. En este caso, las cantidades a producir ya no forman parte del modelo. Para el caso estudiado en Fleischmann (1990), además se consideran opciones de secuenciamiento, donde también se decide el orden en

que se producen los lotes. Otras adaptaciones de este problema regulan la producción, de modo que el nivel de producción puede graduarse sólo a tamaños de lotes discretos.

Problemas multi-nivel y multi-instalación

Los problemas multi-nivel, consideran que los productos están constituidos por componentes. De modo de producir una unidad de un producto, se debe previamente contar con las cantidades de componentes necesarios. De esta manera, mientras que en el DLSP se considera una demanda por un producto externa al sistema, en los problemas multi-nivel se genera además una demanda interna de componentes. En los problemas con múltiples instalaciones, para llegar al producto final terminado, el producto debe pasar por una serie de procesos, que se llevan a cabo en distintas instalaciones. En Zangwill (1969), se estudia un problema con múltiples instalaciones en serie, en el que la producción final de una instalación se vuelve el insumo de la siguiente instalación. Así, la última instalación es la encargada de satisfacer la demanda. En dicho trabajo, se introduce la formulación del problema como flujos en red.

2.1.5. Clasificación de problemas de dimensionado de lotes

En la literatura es posible encontrar distintas clasificaciones para los problemas de dimensionado de lotes. Algunos criterios utilizados se pueden encontrar en Brahimí et al. (2017). Dentro de las clasificaciones que más comúnmente aparecen se destacan:

- **La cantidad de máquinas utilizadas.** En los problemas con múltiples máquinas, se considera la producción de varios productos, que pueden ser producidos en máquinas. Algunos de los productos se pueden asignar exclusivamente a una máquina, mientras que la producción de otros se puede dividir en varias máquinas. Generalmente, la configuración de las máquinas debe ser coordinada, de modo que el problema no se puede descomponer en un subproblema por cada máquina.
- **Característica del lote a producir.** La cantidad a producir puede ser continua o discreta. Un ejemplo de un problema de la realidad donde los lotes sólo pueden tomar tamaño discreto es la industria de alimentos.
- **La cantidad de etapas o niveles.** En el caso de problemas de dimensionado con un único nivel, se considera únicamente demanda externa al sistema. Como se señaló en la sección anterior, también es posible contar con demanda interna. En estos casos, la producción de un ítem genera una demanda dependiente de componentes necesarios para su producción.

- **La existencia o no de capacidades para producir.** Se distingue entre restricciones con capacidades fijas o variables.
- **La duración de los períodos.** Los problemas se pueden dividir acorde a la duración de los períodos. Normalmente se consideran dos categorías: períodos de tiempo cortos (small bucket), del orden de horas, y períodos de tiempos largos (large bucket), del orden de días o semanas. Normalmente, en los problemas large bucket, varios ítems pueden ser producidos en una misma máquina en un mismo período, mientras que para los problemas small bucket, sólo se permite producir un único ítem por período en una máquina.

2.2. Análisis del dimensionado de lotes con remanufactura determinista

La remanufactura de productos atiende las nuevas consideraciones sobre el impacto ambiental de la producción, ya que se tiene el potencial de disminuir la necesidad de consumir recursos naturales, y se reduce la contaminación durante el proceso productivo (Teunter et al., 2006; Cunha y Melo, 2016). En esta sección se presenta un estudio bibliográfico de extensiones del DLSP que incorporan remanufactura, haciendo especial énfasis en el Problema de Dimensionado de Lotes con Remanufactura. Esta extensión permite que la demanda se pueda satisfacer tanto con la producción de artículos nuevos, como a través de un proceso de remanufacturación. El proceso de remanufacturación transforma artículos usados (retornos) que son devueltos al origen, teniendo como resultado artículos que se garantiza al menos la misma calidad y funcionalidad que la de los productos nuevos (Ijomah, 2002). A estos productos se los denomina como productos listos.

El desarrollo de esta sección es la siguiente. En 2.2.1, se presentan posibles agrupaciones para variantes de problemas que incorporan remanufactura. En 2.2.2 se presentan los primeros trabajos sobre problemas con remanufactura. En 2.2.3 se presentan formulaciones para las dos variantes deterministas del caso de estudio del Capítulo 3. En 2.2.4 se presentan los principales resultados del análisis de complejidad de problemas similares al caso de estudio. En relación a estrategias de resolución, las mismas se presentarán al final de esta sección, en la Sección 2.3. Además de las principales estrategias utilizadas para el caso determinista, en dicha sección se presentarán fundamentos teóricos necesarios para una mejor comprensión de las mismas.

2.2.1. Variantes que incorporan remanufactura de retornos

Se presenta a continuación una posible clasificación para problemas de producción de lotes que incorporan remanufacturación.

- Sistemas mixtos de producción de artículos nuevos y remanufacturación: la demanda se puede satisfacer tanto por productos nuevos, como por productos remanufacturados. Un ejemplo de un sistema de este tipo es el Problema de Dimensionado de Lotes con Remanufactura que se verá en la Sección 2.2.3.

- Sistemas de remanufactura puros: el único proceso considerado es el de remanufacturar artículos. Estos sistemas deben tener una cantidad de retornos suficiente para satisfacer la demanda en todos los períodos para ser factibles.
- Sistemas con sustitución: donde se considera que los productos nuevos pueden sustituir a los productos remanufacturados, pero no a la inversa.
- Sistemas de Remanufactura con desmontaje: se modela explícitamente las diferentes etapas del proceso de remanufactura. Una vez devueltos, los retornos se desmontan, obteniéndose un conjunto de partes posiblemente recuperables. Las partes que pueden ser recuperadas se reacondicionan, y pueden ser utilizadas al remanufacturar los artículos.
- Sistemas con disposición final: en estos sistemas los retornos pueden ser desechados por motivos de exceso o calidad. Similarmente, en los sistemas de desmontaje es posible desechar partes recuperables. Nuestro trabajo no considera la posibilidad de desechar artículos. Teunter et al. (2006), señalan que la posibilidad de desechar artículos no representa una mejora considerable en los valores óptimos obtenidos, salvo en casos con cantidad de retornos excesivamente altos.
- Sistemas con calidad de retornos heterogéneos: en estos sistemas además de considerar la cantidad de retornos recibidos, es de interés modelar la calidad de los mismos. Ejemplos de sistemas con estas características se pueden encontrar en Devoto et al. (2021), Mahapatra et al. (2012) y Yanikoğlu y Denizel (2021).

2.2.2. Primeros modelos del problema de dimensionado de lotes con remanufactura

El trabajo de Richter y Sombrutzki (2000) es uno de los primeros estudios que incorpora gestión de inventario con retornos. Los autores estudian un caso especial del problema en el que se asume que la cantidad de artículos retornados en el primer período es suficiente para satisfacer todas las demandas del horizonte de planificación. La producción de artículos nuevos no es indispensable pero se puede utilizar para conseguir mejores valores de función objetivo. El problema es resuelto utilizando un algoritmo de programación dinámica, que es una extensión del conocido algoritmo de Wagner y Whitin (1958). Richter y Weber (2001) incorporan al modelo anterior costos de producción unitarios. Golany et al. (2001) estudian el problema con la opción de desechar artículos. Formulan el problema como un problema de red de flujos y demuestran que el problema es NP-hard cuando se tienen costos con funciones

cóncavas. Yang et al. (2005) extienden el resultado anterior, demostrando que el problema es NP-hard cuando se tienen costos estacionarios cóncavos. Beltrán y Krass (2002) estudian un problema en el que los retornos son considerados al menos tan buenos como los artículos nuevos, y se pueden usar sin remanufacturar.

Teunter et al. (2006) estudian dos variantes del problema en la que no se permite desechar retornos, no se incorporan costos de producción unitarios a la formulación, y tanto los costos de configuración como de inventario son invariantes en el tiempo. En una de las variantes, las actividades de producción de artículos nuevos y remanufacturación son llevadas a cabo en distintas líneas de producción (ELSRs), mientras que en la otra variante la línea de producción es compartida (ELSRj). Ambas formulaciones contienen restricciones de activación implementadas con “M grande”, lo que implica que las formulaciones posiblemente se puedan ajustar haciendo cambios sencillos sobre esas restricciones, como se verá en la siguiente Sección.

Piñeyro y Viera (2009) analizan políticas de inventario específicamente diseñadas para el ELSR con disposición final de retornos. En un trabajo siguiente, Piñeyro y Viera (2010) analizan un problema de remanufactura en el que la demanda de productos remanufacturados se puede satisfacer con nuevos productos, pero no a la inversa. A su vez, se permite la opción de deshechar retornos. Muestran que el problema es NP-hard, incluso para casos particulares del problema, y desarrollan una heurística de tipo Búsqueda Tabú para su resolución.

Existen diversos trabajos que abordan variantes capacitadas de problemas de dimensionado con remanufacturación. Pan et al. (2009) analizan un problema capacitado que incorpora la opción de deshechar productos. Zhang et al. (2012) estudian una variante en la que existen demandas independientes para los productos nuevos y para los remanufacturados. La formulación es resuelta mediante una relajación Lagrangiana. Sahling (2013) estudian la versión capacitada multi-producto, con la opción de deshechar retornos, incluyendo la formulación del problema, y una estrategia de resolución de programación dinámica.

Sifaleras et al. (2015) investigan variantes de la metaheurística de Búsqueda de Entorno Variable (Variable Neighbourhood Search, VNS), para un problema de dimensionado de lotes con remanufactura. También presentan un conjunto de instancias de prueba, que al igual que el presentado en Retel Helmrich et al. (2014), permiten evaluar los modelos que se presentan en la Sección 2.2.3 de este informe. En un trabajo siguiente, Sifaleras y Konstantaras (2017) presentan una metaheurística de Búsqueda de Entorno Variable (Variable Neighbourhood

Search, VNS), para un problema de dimensionado de lotes con remanufactura y la posibilidad de producir múltiples productos. Suzanne et al. (2020) presentan una revisión de los trabajos más importantes relacionados con el concepto de economía circular, y los categorizan según el tipo de operación de recuperación que se lleva a cabo.

En la Sección 2.4 se presentarán otros estudios que consideran algún tipo de incertidumbre en la remanufactura de artículos (Macedo et al., 2016; Naeem et al., 2013). En el resto de esta sección, y en la Sección 2.3 se presenta en mayor detalle trabajos deterministas de problemas similares al caso de estudio del Capítulo 3 (Retel Helmrich et al., 2014; Cunha y Melo, 2016). Dentro de esta revisión, se verá el análisis de complejidad de estudios sobre problemas similares, formulaciones y estrategias de resolución.

2.2.3. Formulación determinista de ELSRs y ELSRj

Una formulación que incorpora costos de producción dinámicos a los problemas con líneas de producción separadas ELSRs (Economic Lot-Sizing with Remanufacturing and Separate setups) y con línea de producción común ELSRj (Economic Lot-Sizing with Remanufacturing and Joint setups) es provista por Retel Helmrich et al. (2014). A continuación se presentan formulaciones que incorporan estos costos para ambos problemas.

2.2.3.1. Definición de conjuntos, parámetros y variables de ELSRs y ELSRj

Se considera un conjunto de períodos T . Para cada período $t \in T$ se tiene una demanda d_t a satisfacer y una determinada cantidad de artículos devueltos r_t . Se identifica el último período como t_f . A su vez, en cada período se considera los siguientes costos:

- h_t^s, h_t^r : costos de almacenar productos listos y de almacenar retornos respectivamente
- K_t^m, K_t^r, K_t : para el problema con líneas de producción separadas, K_t^m y K_t^r son los costos de configuración para la producción de artículos nuevos y para la remanufactura respectivamente. En caso de tratarse del problema con una única línea de producción, se tiene un único costo K_t para ambas actividades
- p_t^m, p_t^r : costos de producción unitarios para la producción de artículos nuevos y para la remanufactura respectivamente

Se tienen dos familias de variables, x_t^m y x_t^r para representar la cantidad de productos nuevos producidos y remanufacturados en t respectivamente. Hay dos esquemas para modelar las variables de activación, dependiendo si se tiene una línea de producción compartida, o líneas

de producción dedicadas. Para el caso con líneas de producción separadas, se tiene variables binarias para activar la producción de artículos nuevos y_t^m y para la remanufactura y_t^r . Para el caso con línea de producción única, solamente se tiene la familia de variables y_t . Finalmente, para cada período se tiene variables para los inventarios de artículos listos I_t^s y para los retornos I_t^r .

2.2.3.2. Formulación ELSRs

Partiendo de la definición de conjuntos, parámetros y variables anterior, la formulación para el ELSRs es:

$$(\text{ELSRs}) : \min \sum_{t \in T} (K_t^m y_t^m + K_t^r y_t^r + h_t^s I_t^s + h_t^r I_t^r + p_t^m x_t^m + p_t^r x_t^r) \quad (2.6)$$

s.a :

$$I_{t-1}^s + x_t^m + x_t^r = I_t^s + d_t, \quad t \in T \quad (2.7)$$

$$I_{t-1}^r + r_t = I_t^r + x_t^r, \quad t \in T \quad (2.8)$$

$$I_0^s = I_0^r = 0, \quad (2.9)$$

$$x_t^m \leq M_t^m y_t^m, \quad t \in T \quad (2.10)$$

$$x_t^r \leq M_t^r y_t^r, \quad t \in T \quad (2.11)$$

$$x_t^m, x_t^r, I_t^s, I_t^r \geq 0, \quad t \in T \quad (2.12)$$

$$y_t^m, y_t^r \in \{0, 1\}, \quad t \in T \quad (2.13)$$

donde $M_t^m = \sum_{i=t}^{t_f} d_i$ y $M_t^r = \min\{\sum_{j=1}^t r_j, \sum_{i=t}^{t_f} d_i\}$.

El objetivo (2.6) es minimizar los costos asociados a los procesos de producir artículos nuevos y remanufacturar. Las restricciones (2.7) indican que la demanda se puede satisfacer con productos nuevos, con productos remanufacturados ó con productos que estaban presentes en el inventario de productos listos al principio del período. A su vez, el excedente pasa al siguiente período como inventario de productos listos. Las restricciones de balance de inventario para retornos, (2.8), indican que la remanufacturación se puede llevar a cabo con retornos de un período o presentes en el inventario del período anterior. El excedente pasa al siguiente período como inventario de retornos. En (2.9) se asigna valores a los inventarios iniciales. Inicialmente tanto los inventarios de productos listos como de retornos están vacíos. Las restricciones (2.10) y (2.11) son de activación de producción. Indican respectivamente que, para poder producir artículos nuevos/remanufacturar en un período t se deben activar la variables y_t^m y y_t^r .

En cuanto a la selección de M_t^m y M_t^r en las restricciones (2.10) y (2.11) puede resultar conveniente elegir como valores los menores posibles. Pochet y Wolsey (2006) indican que las restricciones de “M grande” en los problemas de dimensionado de lotes pueden llevar a relajaciones de programación lineal más débiles para su resolución computacional. Por lo tanto, se elige como valores los menores posibles, dado que así se espera tener restricciones más ajustadas. De esta manera, la formulación será más cercana al casco convexo del problema. En el trabajo de Retel Helmrich et al. (2014), la producción de artículos nuevos y remanufacturación están limitadas por las demandas acumuladas desde el período en cuestión hasta el último período. Sin embargo, la inecuación para la remanufacturación se puede ajustar aún más siguiendo la idea presentada por Cunha y Melo (2016), que hace uso también de los retornos acumulados. De esta manera, la cantidad que se remanufactura en un período está limitada por el mínimo entre los retornos que hay hasta ese período, y la demanda de ese período hasta el final.

Finalmente, (2.12) y (2.13) son restricciones de dominio. Las variables de activación de los procesos de producción en cada período son binarias, mientras que el resto de las variables son continuas y no negativas. Al tratarse de un problema que admite una representación de flujos en red, las variables de producción y de inventario pueden ser definidas como continuas, ya que los valores de las mismas en la solución óptima serán enteros cuando la demanda tome valores enteros (Yang et al., 2005).

2.2.3.3. Formulación ELSRj

Partiendo de la definición de conjuntos, parámetros y variables vista en la Sección anterior, la formulación para el ELSRj es:

$$(\text{ELSRj}) : \min \sum_{t \in T} (K_t y_t + h_t^s I_t^s + h_t^r I_t^r + p_t^m x_t^m + p_t^r x_t^r) \quad (2.14)$$

s.a :

$$I_{t-1}^s + x_t^m + x_t^r = I_t^s + d_t, \quad t \in T \quad (2.15)$$

$$I_{t-1}^r + r_t = I_t^r + x_t^r, \quad t \in T \quad (2.16)$$

$$I_0^s = I_0^r = 0, \quad (2.17)$$

$$x_t^m + x_t^r \leq M_t y_t, \quad t \in T \quad (2.18)$$

$$x_t^m, x_t^r, I_t^s, I_t^r \geq 0, \quad t \in T \quad (2.19)$$

$$y_t \in \{0, 1\}, \quad t \in T \quad (2.20)$$

donde $M_t = \sum_{i=t}^{t_f} d_i$.

El objetivo (2.14) es minimizar los costos asociados a los procesos de producir artículos nuevos y remanufacturar. Las restricciones (2.15) indican que la demanda se puede satisfacer con productos nuevos, con productos remanufacturados ó con productos que estaban presentes en el inventario de productos listos al principio del período, mientras que el excedente pasa al siguiente período como inventario de productos listos. Las restricciones (2.16) son de balance de inventario para los retornos, e indican que la remanufacturación se puede llevar a cabo con retornos de este período o presentes en el inventario del período anterior. El excedente pasa al siguiente período como inventario de retornos. En (2.17) se asigna valores a los inventarios iniciales. Inicialmente tanto los inventarios de productos listos como de retornos están vacíos. (2.18) son las restricciones de activación de la producción, e indican que se debe activar la variable y_t para poder producir artículos nuevos ó remanufacturar.

Para la selección de M_t en las restricciones (2.18) se utiliza una estrategia análoga a la presentada en la Sección 2.2.3.2. Se seleccionan los menores valores M_t posibles para cada período t , que se obtiene de la demanda de ese período hasta el final del horizonte de planificación.

Finalmente, (2.19) y (2.20) son restricciones de dominio. La variable de activación del proceso de producción es binaria en cada período, mientras que el resto de las variables son continuas y no negativas, por un motivo análogo al explicado en la Sección 2.2.3.2.

2.2.4. Complejidad de ELSRj y ELSRs

En esta sección se presentan los principales resultados del análisis de complejidad de problemas similares al caso de estudio del Capítulo 3. En primer lugar se presentan variantes con resolución eficiente, y luego se presentan los resultados correspondientes a las formulaciones ELSRs y ELSRj de la Sección anterior.

Variantes con resolución eficiente

Teunter et al. (2006) estudian dos variantes del problema en el que no se incorporan costos de producción unitarios a la formulación, y tanto los costos de configuración como de inventario son invariantes en el tiempo. Para la formulación con una única línea de producción (ELSRj), desarrollan un algoritmo exacto de programación dinámica de orden $O(|T|^4)$. El algoritmo es una adaptación del método de Wagner y Whitin (1958). La resolución se basa en dos condiciones que necesariamente cumplen las soluciones óptimas:

1. El nivel de inventario del período anterior al período donde se produce (se produce artículos nuevos o remanufacturados) es cero. Esta propiedad se conoce como la propiedad de inventario cero (Zero Inventory Property, ZIP).
2. Hay producción positiva en un período, si la cantidad de retornos disponibles (nivel de inventario del período anterior más los retornos del período) es cero.

Las condiciones anteriormente mencionadas no se cumplen necesariamente cuando se incluyen costos de producción unitarios y los costos varían en el tiempo, como se detalla a continuación.

Complejidad del ELSRs

Baki et al. (2014) y Retel Helmrich et al. (2014) demuestran independientemente que el ELSRs es NP-hard para el caso general con costos de producción unitarios variantes en el tiempo. Para su demostración, se reduce polinomialmente el problema NP-complete conocido como Partición de Garey y Johnson (1990), a la versión del problema de decisión ELSRs. Retel Helmrich et al. (2014) demuestra además que incluso cuando los costos son invariantes en el tiempo, si la demanda es creciente en el tiempo, el problema también es NP-hard.

Complejidad del ELSRj

Retel Helmrich et al. (2014) muestra que cuando se consideran costos de producción unitarios, y costos variantes en el tiempo, el problema es NP-hard. Para su demostración, prueban que dada una instancia del ELSRs, esta siempre puede ser reducida polinomialmente a una instancia del ELSRj donde se duplica la cantidad de períodos. Como el ELSRs es NP-hard, y el ELSRj es al menos tan difícil como resolver el ELSRs, se concluye que el ELSRj es NP-hard.

2.3. Estrategias de resolución para el caso determinista

En esta sección se mencionan estrategias de resolución relevantes para el caso de estudio del Capítulo 3. Como se ha mencionado en secciones anteriores, los problemas ELSRs y ELSRj de las Secciones 2.2.3.2 y 2.2.3.3 son NP-hard en el caso general. Por lo tanto, no se conocen métodos de resolución que puedan determinar soluciones óptimas en tiempo polinomial.

En general, las estrategias de resolución para problemas de optimización se suelen agrupar en dos categorías: métodos de resolución exacta y métodos de resolución heurística. Normalmente existe un compromiso entre la calidad de las soluciones obtenidas con respecto a los óptimos de los problemas, y los tiempos de cómputo para obtener dichas soluciones. Los métodos de resolución exacta, garantizan que en caso de que exista una solución óptima, se llegará a la misma si se brinda al método los recursos computacionales suficientes y el tiempo necesario de ejecución. A partir de determinados tamaños de instancia, la capacidad de cómputo necesaria puede ser muy superior a la disponible. En estos casos, los métodos exactos generalmente pueden brindar una cota de calidad de la solución obtenida, al ejecutarse con algún criterio de parada. Los métodos de resolución heurística, en contrapartida, requieren una capacidad de cómputo sensiblemente menor a las estrategias de resolución exacta, pero habitualmente no dan garantías de encontrar soluciones cercanas a las óptimas.

Para poder aplicar métodos de resolución exacta a instancias de un problema NP-hard de mayor tamaño, normalmente se incorporan criterios de parada específicos para disminuir los tiempos de cómputo. Maes y Wassenhove (1988) se refieren a estos métodos como heurísticas basadas en programación matemática. Dentro de estas estrategias basadas en métodos de resolución exacta, se presenta en primer lugar los métodos de branch-and-bound en la Sección 2.3.1. Estos métodos son el mecanismo por excelencia utilizado por los paquetes informáticos de optimización (solver) para resolver los problemas de programación matemática. En la Sección 2.3.2, se verán las principales formulaciones extendidas utilizadas en la literatura para problemas con remanufactura. En la Sección 2.3.3 se verán los métodos de planos de corte a partir de la incorporación de inecuaciones válidas.

Por otro lado, existen otras heurísticas que se rigen por el uso de reglas específicas y no por la utilización de técnicas matemáticas. Para el desarrollo de alguna de estas heurísticas, se intenta explotar aspectos operativos específicos del problema, como pueden ser propiedades de los lotes, de la demanda o de los inventarios. Una mayor descripción de este tipo de heurísticas se puede encontrar en Maes y Wassenhove (1988) y ejemplos de su aplicación en

Piñeyro y Viera (2010), Baki et al. (2014), Chen y Abrishami (2014), Koken et al. (2018), Zouadi et al. (2018), entre otros. En la Sección 2.3.5.1 se presenta en particular la heurística relax-and-fix, la cual sirve de base para el desarrollo del Capítulo 5.

Si bien a lo largo de este capítulo se mencionan los principales resultados teóricos de estrategias de resolución MIP, se sugiere Pochet y Wolsey (2006) para una comprensión más exhaustiva de conceptos básicos de los problemas MIP.

2.3.1. Ramificado y acotamiento

El método de Ramificado y Acotamiento (branch-and-bound), es un método que genera un árbol de problemas, y utiliza cotas del valor de la función objetivo. La eficiencia del mismo depende en gran medida del nivel de ajuste que estas cotas provean. El espacio de soluciones factibles se divide para obtener subproblemas de menor tamaño, generándose el árbol de problemas. A medida que se van generando subproblemas, se utilizan reglas en base a las cotas del valor objetivo que se tienen hasta el momento, y así se logra podar ramas del árbol, es decir, se evita resolver subproblemas.

El esquema resuelve en primer lugar la relajación a programación lineal del problema original (nodo inicial) a través de algún método de resolución eficiente. En caso de obtenerse una solución que satisface las restricciones de integralidad, se ha encontrado la solución óptima. En otro caso, se elige alguna de las variables fraccionarias obtenidas en la solución, y se generan dos nuevos problemas: uno donde se exige que la variable seleccionada tome un valor menor o igual que la parte entera obtenida, y otro donde se exige la variable tome un valor mayor o igual que el resultado de sumarle uno a la parte entera. Se generan así dos nuevos subproblemas a resolver con espacios de soluciones factibles más acotados con respecto al del problema original.

Para cada uno de estos subproblemas se resuelven sus relajaciones a programación lineal, y se decide en base a la característica de la solución obtenida:

- si la solución es entera, entonces es óptima del subproblema, por lo que no es necesario seguir ramificando y se poda por optimalidad
- si la solución no es mejor que la mejor solución factible obtenida hasta al momento durante la ejecución del algoritmo, se poda por acotamiento
- si no hay solución factible, se poda por infactibilidad.

- en caso contrario se vuelve a ramificar.

El procedimiento continúa recursivamente mientras existan subproblemas sin procesar. Si no quedan problemas activos, la mejor solución factible obtenida hasta al momento es la solución óptima.

2.3.2. Formulaciones extendidas

En esta sección se presentan las principales formulaciones extendidas que se han desarrollado para el caso determinista del Problema de Dimensionado de Lotes con Remanufactura.

Una formulación extendida de un conjunto factible X de un problema de programación lineal, es una formulación que incorpora nuevas variables en un espacio de mayor dimensión. Con la utilización de formulaciones extendidas se busca aproximarse al casco convexo de X incorporando una cantidad de restricciones que se espera sea menor que con formulaciones cuyas variables están en el espacio original. Formalmente, para definir las formulaciones extendidas, se debe primero definir la proyección de un conjunto sobre una variable. Las definiciones son las siguientes:

Definición 2 (Formulación) Dado $X = \{(x, y) \in \mathbb{R}^n \times \mathbb{Z}^p : Ax + Bw \leq b\}$, el conjunto $P \subseteq \mathbb{R}^{n+p}$ es una formulación para X sii $X = P \cap (\mathbb{R}^n \times \mathbb{Z}^p)$

Definición 3 (Proyección de un conjunto sobre una variable) Dado $Z = \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^p : Dx + Ew \leq c\}$, la proyección de Z sobre x se define como $proj_x(Z) = \{x \in \mathbb{R}^n : \exists w \in \mathbb{R}^p \text{ con } (x, w) \in Z\}$.

Definición 4 (Proyección de un conjunto sobre una variable) Dado $Z = \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^p : Dx + Ew \leq c\}$, la proyección de Z sobre x se define como $proj_x(Z) = \{x \in \mathbb{R}^n : \exists w \in \mathbb{R}^p \text{ con } (x, w) \in Z\}$.

Definición 5 (Formulación extendida) Dado $X = \{x \in \mathbb{Z}^n : Ax \leq b\}$ y un conjunto $Z = \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^p : Dx + Ew \leq c\}$, se dice que Z es una formulación extendida de X cuando $proj_x(Z)$ es una formulación de X .

Las principales reformulaciones que se han desarrollado en la literatura para los problemas con remanufactura, son las denominadas Camino Mínimo, y las de Localización de Instalaciones. En las siguientes subsecciones, se explica en detalle las reformulaciones, y luego se mencionan alguno de los estudios más importantes en relación a las mismas.

2.3.2.1. Camino mínimo

Los problemas de Camino Mínimo (Shortest Path) consisten en encontrar un camino de costo mínimo entre dos vértices de un grafo con costos en sus aristas. Una aplicación posible es el problema de flujos en red en el que una unidad debe ser trasladada desde un vértice origen a un vértice destino de un grafo dirigido. La primer formulación basada en flujos en red para problemas de dimensionado de lotes, fue la propuesta por Eppen y Martin (1987) para el Problema de Dimensionado de Lotes Capacitado. En cada nodo se debe cumplir la propiedad de conservación de flujo, en el que el flujo de entrada al nodo debe coincidir con el flujo de salida.

Este problema tiene una representación que es de utilidad para reformular el problema de lotes clásico. Partiendo de la formulación del DLSP de la Sección 2.1.1, se incorporan variables z_{ij} que representan el porcentaje de la demanda entre los períodos i y j que será atendida con producción generada en el período i . Se parte de un flujo inicial de una unidad, y el flujo que recorre una arista entre dos nodos está dado por el valor de la variable z_{ij} . La producción en un período i se puede obtener a partir de los valores de las aristas salientes desde i hacia nodos t con la siguiente expresión: $x_i = \sum_{t=i}^{t_f} D_{it} z_{it}$, donde D_{it} es la demanda acumulada entre i y t , $D_{i,t} = \sum_{\hat{t}=i}^t d_{i\hat{t}}$. Utilizando esta expresión, resulta sencillo calcular los costos para cada período.

El problema con remanufacturación también puede ser formulado con esta estrategia. En este caso la representación se puede hacer con dos flujos en red (Retel Helmrich et al., 2014; Cunha y Melo, 2016), que se vinculan con la incorporación de nuevas restricciones:

- En una red de flujo, se consideran los productos listos. Para el ELSRs se tienen dos familias de variables z_{ij}^{sm} y z_{ij}^{sr} que determinan la proporción de la demanda de los períodos entre i y j que será atendida con productos nuevos o remanufacturados en el período i respectivamente. Para el ELSRj las dos familias de variables anteriores se pueden agrupar en una única familia, de manera análoga a lo que sucede con las variables de activación de producción para ELSRs y ELSRj.
- En la otra red de flujo, se consideran los productos retornados. Se tiene una familia de variables z_{ij}^r que representa la cantidad de retornos entre i y j que son remanufacturados en j .

Ambos flujos en red se ven vinculados con el agregado de restricciones. Para el ELSRs se debe incorporar las siguientes restricciones: $\sum_{i=1}^t R_{it} z_{it}^r = \sum_{j=t}^{t_f} D_{tj} z_{tj}^{sr}$, con $t \in T$, donde D_{tj} es la demanda acumulada entre t y j , y R_{it} es la cantidad de retornos acumulados entre los

períodos i y t . De esta manera, quedan vinculadas las variables asociadas a los productos listos y los retornos.

A continuación, se presentan los antecedentes más relevantes en los que se estudian formulaciones de este tipo. Eppen y Martin (1987) proponen una formulación de flujos en red para el Problema de Dimensionado de Lotes Capacitado. Retel Helmrich et al. (2014) y luego Cunha y Melo (2016) utilizan esta formulación como base para comparar el comportamiento del ELSRs y ELSRj frente a otras reformulaciones. Cuando el número de períodos es grande y los costos de configuración bajos, es esperable que muchas variables puedan tomar el valor cero. Para reducir el número de variables, Retel Helmrich et al. (2014) realiza una formulación de flujos en red aproximada para el ELSRs basada en el trabajo de Pochet y Wolsey (2006). Dados dos períodos i y j , se elige un parámetro k , de modo que las variables z_{ij} sólo se calculan cuando $i < j + 1 < i + k$. El resto de las variables se agrupan en una nueva variable. Las restricciones de la formulación natural se deben mantener para asegurar la correctitud de la formulación, ya que en caso contrario se podría estar agrandando el espacio de soluciones factibles. La dificultad radica en la selección del valor del parámetro: valores pequeños pueden deteriorar las cotas de la relajación, mientras que valores grandes pueden no ser suficientemente buenos para disminuir el número de variables.

2.3.2.2. Localización de instalaciones

En las reformulaciones de Localización de Instalaciones se desagregan las variables de producción definiendo nuevas variables para estas cantidades que informan de modo más detallado el esquema de producción. Como se explicará en el Capítulo 4, estas reformulaciones pueden ajustar las restricciones de activación de producción, presentes tanto en el problema clásico (DLSP), como en las variantes que incorporan remanufactura. Para el problema clásico, se tendrán variables que indican no sólo dónde se produce sino también cuándo será consumida dicha producción. Para el problema con remanufactura, se incorporan también variables que indiquen de cual período provienen los retornos con los que se está remanufacturando.

Una posible descripción de la redefinición de las variables es la siguiente. Se descomponen las variables de producción de artículos nuevos y remanufacturación de la formulación de la Sección 2.2.3.3. Dado un período i y un período futuro j , se incluyen las siguientes variables:

- cantidad de productos remanufacturados en el período i para satisfacer la demanda en j
- cantidad de productos nuevos en el período i para satisfacer la demanda en j

- cantidad de productos remanufacturados en el período j con productos que fueron retornados en i

Las Reformulaciones de Localización de Instalaciones fueron originalmente propuestas por Bilde y Krarup (1977) para resolver el problema clásico (DLSP). Desde entonces, se han aplicado con éxito para distintas variantes del problema, como se puede estudiar en Pochet y Wolsey (2006). Cunha y Melo (2016) y luego Ali (2017) estudian reformulaciones de este tipo para los problemas ELSRj y ELSRs. En el Capítulo 4 se presenta una reformulación de Localización de Instalaciones, para las versiones que incorporan incertidumbre en las demandas y retornos para los problemas ELSRj y ELSRs.

2.3.3. Planos de corte

Una de las maneras de mejorar el rendimiento de las estrategias de resolución de ramificado y acotamiento (branch-and-bound) es la incorporación de inecuaciones válidas en la formulación, también conocidas como Planos de Corte.

Definición 6 (Inecuaciones válidas) *Dado un conjunto X y una inecuación de la forma $\pi^T x \leq \pi_0$, se dice que esta inecuación es válida para X si la inecuación se cumple para todo $x \in X$.*

La incorporación de inecuaciones válidas permite tener una formulación más ajustada que potencialmente genere cotas más próximas al óptimo durante el algoritmo de ramificado y acotamiento. Dado un problema de programación lineal entera $\min_x \{c^T x \mid x \in X \subset \mathbb{Z}^n\}$ se busca una descripción lo más cercana posible al casco convexo de las soluciones factibles $\text{conv}(X)$. En caso de conocerse una descripción del casco convexo, la resolución del problema coincide con la resolución del problema de programación lineal sin restricciones de integralidad $\min_x \{c^T x \mid x \in \text{conv}(X)\}$. Sin embargo, para muchos problemas no se conoce una descripción completa del casco.

Mediante un proceso de inclusión de inecuaciones se intenta ajustar la formulación del problema. Dado que la cantidad de inecuaciones válidas generadas puede ser muy elevada, la dificultad radica en encontrar inecuaciones relevantes, que permitan reducir de mejor manera el espacio de soluciones y obtener una formulación que sea lo más próxima al casco convexo.

Como fue mencionado anteriormente, el problema DLSP tiene resolución eficiente. Cuando un problema se puede resolver eficientemente, es posible realizar una descripción del casco convexo con una cantidad polinomial de inecuaciones válidas (Pochet y Wolsey, 2006). En la

siguiente sección se presentan inecuaciones válidas que permiten describir el casco convexo de dicho problema.

2.3.3.1. Inecuaciones válidas para el DLSP

Barany et al. (1984a) demuestran que el casco convexo se puede describir completamente con la utilización de restricciones básicas de la formulación natural y la inclusión de inecuaciones válidas de tipo (l, S) . A continuación se presentan las inecuaciones válidas propuestas en orden cronológico.

Proposición 1 (Barany et al., 1984a) *Sea $l \in T$ y $S \subseteq L = \{1, \dots, l\}$, la siguiente familia de inecuaciones, denominada (l, S) , es válida para el problema DLSP:*

$$\sum_{i \in S} x_i + \sum_{i \in L \setminus S} D_{il} y_i \geq D_{1l} \quad (2.21)$$

Equivalentemente, la inecuación se puede escribir como en Barany et al. (1984b) agregando las variables de inventario:

$$\sum_{i \in S} x_i \leq \sum_{i \in S} D_{il} y_i + I_l \quad (2.22)$$

Desde un punto de vista computacional, dado que la cantidad de inecuaciones (2.21) y (2.22) es exponencial en la cantidad de períodos, resulta poco conveniente agregar todas desde un principio. Ante esta situación, una de las posibilidades es procurar fortalecer la formulación seleccionando un subconjunto de esta familia de inecuaciones. Sin embargo, se presenta una nueva dificultad que radica en encontrar las inecuaciones que realmente son convenientes. Una posibilidad, es utilizar un algoritmo de separación como en Barany et al. (1984b).

Cuando se consideran costos de Wagner-Within (ver Sección 2.1.3), el casco convexo se puede obtener sin necesidad de utilizar todos los posibles subconjuntos S en las inecuaciones válidas (2.21). Como fue estudiado en Pochet y Wolsey (1994), para el problema con este tipo de costos, la utilización de un subconjunto de las inecuaciones (l, S) , que se denominarán como (l, S, WW) es suficiente para obtener el casco convexo de las soluciones factibles. En la expresión (2.23) se presenta este subconjunto de inecuaciones. Esta expresión se diferencia de la expresión (2.21) en que, dado un período l , en lugar de tomar un conjunto cualquiera $S \subseteq \{1, \dots, l\}$ para generar la inecuación, solamente se pueden tomar conjuntos que toman un rango de valores desde el primer período.

$$\sum_{i=1}^{k-1} x_i + \sum_{i=k}^l D_{il} y_i \geq D_{1l}, \quad 1 \leq k \leq l \leq |T| \quad (2.23)$$

Equivalentemente, la inecuación se puede escribir agregando variables de inventario como:

$$I_{k-1} + \sum_{i=k}^l D_{il} y_i \geq D_{kl} \quad 1 \leq k \leq l \leq |T| \quad (2.24)$$

En este caso, puede no ser necesario contar con un algoritmo de separación, ya que la cantidad de inecuaciones a agregar es polinomial en la cantidad de períodos. De todos modos, un algoritmo de separación para las inecuaciones de este tipo se puede encontrar en Pochet y Wolsey (2006)

A diferencia del problema clásico DLSP, los problemas ELSRs y ELSRj son NP-hard. Con la intención de obtener formulaciones más proximas al casco convexo del problema, se han adaptado las inecuaciones válidas anteriores al problema con remanufactura. En la siguiente Sección se presentan distintas versiones de estas inecuaciones que han sido estudiadas por varios autores.

2.3.3.2. Inecuaciones válidas para el problema con remanufactura determinista

En esta sección se exponen inecuaciones válidas para el problema con remanufactura determinista. Para su presentación, se representa con R_{ij} la cantidad de retornos acumulados entre los períodos i y j , y con D_{ij} la demanda acumulada entre i y j .

Inecuaciones (l, S, WW) para problema con remanufactura

Las inecuaciones (l, S, WW) propuestas por Pochet y Wolsey (1994) fueron adaptadas por Retel Helmrich et al. (2014) para el ELSRs y ELSRj, derivándose dos inecuaciones validas para cada problema. Los resultados se presentan en las siguientes dos proposiciones:

Proposición 2 (Retel Helmrich et al., 2014) *Dados los períodos i, j , tales que $2 \leq i \leq j \leq |T|$. Las siguientes inecuaciones son válidas para el ELSRs:*

$$I_{i-1}^S + \sum_{t=i}^j D_{tj} (y_t^m + y_t^r) \geq D_{ij}, \quad 2 \leq i \leq j \leq |T| \quad (2.25)$$

$$I_j^r + \sum_{t=i}^j R_{it} y_t^r \geq R_{ij}, \quad 1 \leq i \leq j \leq |T| \quad (2.26)$$

Proposición 3 (Retel Helmrich et al., 2014) *Dados los período i, j , tales que $2 \leq i \leq j \leq |T|$. Las siguientes inecuaciones son válidas para el ELSRj:*

$$I_{i-1}^s + \sum_{t=i}^j D_{tj} y_t \geq D_{ij}, \quad 2 \leq i \leq j \leq |T| \quad (2.27)$$

$$I_j^r + \sum_{t=i}^j R_{it} y_t \geq R_{ij}, \quad 1 \leq i \leq j \leq |T| \quad (2.28)$$

Las inecuaciones sobre la demanda presentadas anteriormente no consideran el caso borde del primer período. Ali (2017) toma en cuenta estas inecuaciones propuestas, y agrega a las inecuaciones (2.25) y (2.27) las inecuaciones que consideran el primer período. Es decir, sin necesidad de modificar estas familias, se pueden agregar inecuaciones de modo de tener inecuaciones válidas para $1 \leq i \leq j \leq |T|$.

Para que una solución del ELSRs sea óptima, la cantidad remanufacturada en un período no debe ser mayor a las demandas acumuladas desde ese período hasta el último de los períodos. Asimismo, la cantidad remanufacturada no puede superar la cantidad de retornos que se recibieron desde el primer período hasta el período en consideración. Por lo tanto, la remanufactura no puede superar el mínimo de las dos cantidades mencionadas anteriormente. Sea R_{1t} la cantidad de retornos acumulada entre el primer período y un período t . Entonces, la familia de inecuaciones (2.25) se puede fortalecer siguiendo la idea desarrollada por Cunha y Melo (2016) :

$$I_{i-1}^s + \sum_{t=i}^j D_{tj} y_t^m + \sum_{t=i}^j \min(R_{1t}, D_{tj}) y_t^r \geq D_{ij}, \quad 2 \leq i \leq j \leq |T| \quad (2.29)$$

Ali (2017) agrega dos nuevas familias de inecuaciones (l, S, WW), (2.30) y (2.31), para el ELSRs a las inecuaciones anteriormente propuestas (2.25) y (2.26). En nuestro estudio experimental se comprobó que se obtienen mejoras en las cotas obtenidas a través de la relajación a programación lineal cuando estas inecuaciones se incorporan a la formulación con inecuaciones válidas (2.25) y (2.26).

Proposición 4 (Ali, 2017) *Dados los período k, l , tales que $1 \leq k \leq l \leq |T|$. Las siguientes inecuaciones son válidas para el ELSRs:*

$$I_{k-1}^s + \sum_{i=k}^l D_{il} y_i^r + \sum_{i=k}^l (I_i^s - I_{i-1}^s - I_{i-1}^r - r_i + I_i^r + d_i) \geq D_{kl} \quad 1 \leq k \leq l \leq |T| \quad (2.30)$$

$$I_{k-1}^s + \sum_{i=k}^p D_{il} y_i^m + \sum_{i=k}^p (r_i + I_{i-1}^r - I_i^r) \geq D_{kl} \quad 1 \leq k \leq l \leq |T| \quad (2.31)$$

Cunha y Melo (2016) describen una nueva familia de inecuaciones válidas partiendo de la definición de parámetros auxiliares rr_k y \underline{d}^p :

- rr_k : cantidad residual de retornos que habría en el período k si se tuviera un esquema de producción en el que siempre se utilizaran retornos para satisfacer la demanda cuando haya disponibilidad de los mismos. Es decir, la cantidad de retornos con los que se cuenta al principio de k , antes de satisfacer la demanda de dicho período, si en los períodos anteriores las demandas fueron satisfechas con retornos siempre que fue posible. Así, en el período siguiente $k + 1$, la cantidad de retornos será la suma de los retornos en $k + 1$, más los retornos que hayan quedado del período anterior. El valor del parámetro se puede obtener recurivamente de la siguiente manera,

$$rr_k = \begin{cases} r_k + \max(0, rr_{k-1} - d_{k-1}), & \text{si } 2 \leq k \leq |T|, \\ r_k, & \text{si } k = 1. \end{cases}$$

- \underline{d}^p : demanda mínima que necesariamente debe ser satisfecha con producción de artículos nuevos, dada la disponibilidad de artículos retornados,

$$\underline{d}_k^p = \max(0, d_k - rr_k)$$

En base a la definición anterior, se define las demandas mínimas acumuladas desde el primer período hasta un período k como $\underline{d}_{1k}^p = \sum_{l=1}^k \underline{d}_l^p$.

Teniendo en cuenta las definiciones mencionadas, se genera una nueva inecuación válida de tipo (l, S, WW) para el parámetro \underline{d}^p , como se muestra en la siguiente proposición.

Proposición 5 (Cunha y Melo, 2016) *Dados los período t, l , tales que $1 \leq t \leq l \leq |T|$. Las siguientes inecuaciones son válidas para el ELSRs:*

$$\sum_{k=1}^{t-1} x_k^m + \sum_{k=t}^l \underline{d}_{kl}^p y_k^p \geq \underline{d}_{1l}^p \quad 1 \leq t \leq l \leq |T| \quad (2.32)$$

Como fue visto anteriormente, las inecuaciones (l, S, WW) son un subconjunto de las inecuaciones (l, S) propuestas por Barany et al. (1984a) y Barany et al. (1984b). A continuación, se presentan las versiones (l, S) para los problemas ELSRs y ELSRj.

Inecuaciones (l, S) para problema con remanufactura

Ali (2017) propone cuatro familias de inecuaciones válidas para el ELSRs, y dos para el ELSRj adaptadas de las inecuaciones (l, S) de Barany et al. (1984a).

Proposición 6 (Ali, 2017) *Dados los período i, j , tales que $1 \leq i \leq j \leq |T|$ y conjuntos L y S , $L = \{k, \dots, l\}$, $S \subseteq L$. Las siguientes inecuaciones son válidas para el ELSRs:*

$$\sum_{i \in S} x_i^r \leq \sum_{i \in S} R_{ki} y_i^r + I_{k-1}^r \quad 1 \leq i \leq j \leq |T|, L = \{k, \dots, l\}, S \subseteq L \quad (2.33)$$

$$\sum_{i \in S} (x_i^r + x_i^m) \leq \sum_{i \in S} D_{il} (y_i^r + y_i^m) + I_l^s \quad 1 \leq i \leq j \leq |T|, L = \{k, \dots, l\}, S \subseteq L, \quad (2.34)$$

$$\sum_{i \in S} x_i^r \leq \sum_{i \in S} D_{il} y_i^r + I_p^s \quad 1 \leq i \leq j \leq |T|, L = \{k, \dots, l\}, S \subseteq L, \quad (2.35)$$

$$\sum_{i \in S} x_i^m \leq \sum_{i \in S} D_{il} y_i^m + I_p^s \quad 1 \leq i \leq j \leq |T|, L = \{k, \dots, l\}, S \subseteq L, \quad (2.36)$$

Proposición 7 (Ali, 2017) *Dados los período i, j , tales que $1 \leq i \leq j \leq |T|$ y conjuntos L y S , $L = \{k, \dots, l\}$, $S \subseteq L$. Las siguientes inecuaciones son válidas para el ELSRj:*

$$\sum_{i \in S} x_i^r \leq \sum_{i \in S} R_{ki} y_i + I_{k-1}^r \quad 1 \leq i \leq j \leq |T|, L = \{k, \dots, l\}, S \subseteq L, \quad (2.37)$$

$$\sum_{i \in S} (x_i^r + x_i^m) \leq \sum_{i \in S} D_{il} y_i + I_l^s \quad 1 \leq i \leq j \leq |T|, L = \{k, \dots, l\}, S \subseteq L, \quad (2.38)$$

Como fue mencionado en la Sección 2.3.3.1, la incorporación de familias de inecuaciones (l, S) significa la incorporación de una cantidad exponencial de restricciones. Existen principalmente dos enfoques para trabajar con estas familias de inecuaciones. Una posibilidad, es utilizar algoritmos de separación, como los propuestos por Barany et al. (1984a) y luego adaptados para el problema con remanufactura por Ali (2017). Una segunda posibilidad, es utilizar un subconjunto de las restricciones (l, S) , como puede ser las restricciones (l, S, WW) . Esta familia de restricciones es suficiente para obtener el casco convexo de las soluciones factibles del problema clásico con costos de Wagner-Within (Pochet y Wolsey, 1994). Aunque dichas inecuaciones no son suficientes para obtener el casco convexo del ELSRs y ELSRj, las mismas pueden ser utilizadas para el problema con remanufactura, de modo de limitar la cantidad de inecuaciones que se incorporan, enfoque utilizado por varios autores e.g. (Retel Helmrich et al., 2014; Cunha y Melo, 2016).

2.3.4. Equivalencias entre formulaciones

En esta sección se presentan equivalencias teóricas entre las formulaciones de localización de instalaciones, las de camino mínimo, y las que incorporan inecuaciones de tipo (I, S) .

Para el ELSRj, Ali (2017) muestra que la reformulación de localización de instalaciones, la reformulación camino mínimo y la reformulación que incorpora inecuaciones de tipo (I, S) son equivalentes en términos de las cotas inferiores que se obtienen resolviendo el problema relajado a programación lineal.

Para el ELSRs, Ali (2017) muestra que, en este caso, únicamente las reformulaciones de localización de instalaciones y de camino mínimo son equivalentes, difiriendo ambas de la que incorpora inecuaciones de tipo (I, S) . Por otro lado, pese a no contarse con resultados teóricos, en el estudio experimental realizado por el autor, las cotas inferiores obtenidas con las inecuaciones (I, S) fueron de valor igual o superior a las obtenidas utilizando la formulación de localización de instalaciones.

2.3.5. Heurísticas

Los métodos de resolución heurísticos requieren normalmente una capacidad de cómputo sensiblemente menor a las estrategias de resolución exacta. En contrapartida, habitualmente no dan garantías de encontrar soluciones de buena calidad.

Las heurísticas no basadas en programación matemática aplicadas a problemas de dimensionado de lotes, fueron agrupadas por Maes y Wassenhove (1988) en base a dos categorías: heurísticas que construyen la solución avanzando en los períodos y heurísticas de mejora de una solución dada. Las heurísticas de construcción período a período construyen una solución en una única corrida, recorriendo el problema desde el primer período hasta el último. Eventualmente, se pueden realizar corridas siguientes para mejorar la solución. Las heurísticas de mejora de solución parten de una solución, que no necesariamente debe ser factible. Luego, iterativamente se realizan ajustes en la vecindad de la solución buscando o bien obtener factibilidad o bien mejorar la calidad de la solución. Una clasificación más reciente para las heurísticas puede ser encontrada en Brahimí et al. (2017). En dicho trabajo, se mencionan distintas heurísticas para problemas de dimensionado de lotes: heurísticas que utilizan programación matemática (basadas en relajaciones, programación lineal o adaptaciones del método branch-and-bound), heurísticas simples, métodos de aproximación y

metaheurísticas.

En relación a heurísticas aplicadas a problemas con remanufactura, se destacan los siguientes estudios. Teunter et al. (2006) analizan el desempeño de las heurísticas Silver Meal, Least Unit Cost y Part Period Balancing sobre las variantes ELSRj y ELSRs sin costos de producción unitarios. Para ambos casos, los resultados demuestran una superioridad de las dos primeras heurísticas por sobre Part Period Balancing. Van den Heuvel (2006) aplica algoritmos genéticos y programación dinámica para resolver versiones simplificadas del ELSRs en las que los períodos donde se activa la producción son conocidos. Piñeyro y Viera (2009) diseñan una heurística de búsqueda Tabú bajo el principio de “dividir y conquistar” para una variante con remanufactura y la opción de deshechar retornos. Piñeyro y Viera (2010) evalúan una heurística de búsqueda Tabú en el que la demanda de productos remanufacturados se puede satisfacer con nuevos productos, pero no a la inversa. A su vez, se permite la opción de deshechar retornos. Schulz (2011) resuelven el ELSRs con una modificación de la heurística Silver-Meal anteriormente utilizada por Teunter et al. (2006). Sifaleras et al. (2015) investigan variantes de la metaheurística de Búsqueda de Entorno Variable (Variable Neighbourhood Search, VNS), para un problema de dimensionado de lotes con remanufactura.

Para problemas con remanufactura que incorporan restricciones de capacidad, se han sugerido distintas heurísticas. Sahling (2013) estudia la versión capacitada multi-producto, con selección de vendedores. La misma es resuelta con una heurística que combina generación de columnas con la heurística fix-and-optimize. Roshani et al. (2016) utilizan simulated annealing para resolver una versión capacitada en la que la demanda de productos solo se puede satisfacer con productos remanufacturados. Roshani et al. (2017) extienden el problema anterior para aceptar también la producción de productos nuevos y resuelven el problema con una heurística relax-and-fix.

En este trabajo, se aplican ideas del método relax-and-fix como estrategia de resolución del caso de estudio de la Sección 3. A continuación se presentan las ideas básicas del método para el caso determinista en más detalle.

2.3.5.1. Método relax-and-fix

La estrategia detrás de las heurísticas relax-and-fix para problemas MIP es resolver iterativamente, con un solver de optimización tradicional, una serie de problemas de menor tamaño. Algunos ejemplos de este método son provistos por Roshani et al. (2017), Alonso et al. (2000) y Beraldi et al. (2006). Durante el método, se relajan y/o fijan variables enteras/binarias, de

modo que se reduce la cantidad de las mismas. De esta manera, se espera que los problemas resultantes sean más sencillos de resolver dado que el árbol del branch-and-bound, generado internamente por el solver, debe analizar una menor cantidad de variables. En contrapartida, dado que se fija el valor de un subconjunto de variables enteras en cada iteración, en el procedimiento se pierden soluciones factibles del problema original, alguna de las cuales puede ser óptima.

En términos generales, dado el conjunto de variables enteras/binarias del problema original, se debe seleccionar en cada iteración un subconjunto de las mismas a relajarse, otro subconjunto se fija con resultados de iteraciones anteriores, y en otro se mantiene el dominio de la variable del problema original. Existen diversas maneras de seleccionar los subconjuntos en cada iteración. El horizonte de planificación se suele dividir en tres ventanas, inicial, central y final. En la ventana inicial, el valor de las variables binarias se fijan al valor que tomaron en una iteración previa. En la ventana central, se mantiene el dominio de las variables del problema original. En la ventana final, las variables se relajan en dominio a su casco convexo. Para esta última ventana, en caso de tratarse de variables binarias, las variables pueden tomar cualquier valor real entre 0 y 1. En cada iteración, la ventana central avanza a lo largo del horizonte de planificación acorde a determinado criterio, y las ventanas inicial y final se ajustan en consecuencia. El procedimiento continúa de esta manera hasta tanto se llegue a una iteración en la que solo existen las ventanas inicial y media, cuya solución será devuelta por el método.

2.4. Análisis de variantes del dimensionado de lotes con remanufactura estocástico

Todos los modelos presentados en las secciones de este capítulo hasta este punto tienen el supuesto en común de trabajar con parámetros deterministas. En muchos casos de la realidad, no resulta posible conocer o estimar con seguridad el valor que tomarán algunos de los parámetros. Asumiendo que los parámetros con incertidumbre siguen un proceso estocástico de tiempo discreto con espacio de probabilidad finito, el valor de los mismos será determinado luego de conocer la realización de eventos aleatorios en determinado momento del tiempo. De esta manera, el proceso de decisión se llevará a cabo en determinados momentos, o etapas. Para el caso específico de Problemas de Dimensionado de Lotes, generalmente las etapas están dadas por períodos de tiempo pertenecientes al horizonte de planificación. Mientras que en los modelos con incertidumbre de una única etapa se busca crear planes de producción para todo el horizonte de planificación de antemano, los modelos de mayor cantidad de etapas permiten posponer las decisiones a períodos siguientes una vez que se resuelva parte de la incertidumbre. Para el problema con remanufactura, toman particular importancia la incertidumbre de la demanda y la cantidad de retornos. Esto se debe a que muchas veces resulta difícil prever qué cantidad de artículos serán devueltos por los clientes, y a la dificultad de estimar la demanda de artículos listos. En nuestro trabajo, estudiaremos variantes de los problemas ELSRs y ELSRj partiendo del modelado de los mismos mediante programación entera-mixta estocástica, que incorpora al modelado MIP tradicional, la representación de la estructura de la información con incertidumbre mediante un árbol de escenarios.

En el modelado de algunos problemas estocásticos es necesario indicar de alguna manera la etapa en la que se encuentra una variable de decisión. La representación de los índices de la etapa de las variables, utilizando la programación entera-mixta estocástica, se puede abordar a través de dos técnicas. En una de ellas, la etapa de la variable se representa explicitando las realizaciones de los eventos aleatorios que fueron necesarias para llegar a ese estado de situación de la realidad. A modo de ejemplo, para representar una tercer etapa, es necesario explicitar la realización de dos eventos aleatorios previos. En la otra representación, conocida como modelo de escenarios separados, se identifica cada escenario con un identificador único, y se utiliza ese identificador en los subíndices de las variables. Dado que en esta segunda representación, una variable de alguna etapa puede pertenecer a más de un escenario, la misma variable será representada de más de una manera, con distintos subíndices. Por lo tanto, resulta necesario incorporar restricciones que obliguen a estas representaciones de la variable a tener el mismo valor, conocidas como restricciones de no anticipatividad. Estas

restricciones son las únicas que vinculan los escenarios. Una explicación más detallada de estos y otros conceptos básicos de programación entera-mixta estocástica, son provistos por Birge y Louveaux (2011). La representación seleccionada para este trabajo es la de árboles de escenarios, que es utilizada habitualmente para los problemas de dimensionado de lotes. Dicha estrategia se puede clasificar en la primer categoría descrita anteriormente, y su formulación general se verá en la siguiente sección.

El resto de esta sección está organizada de la siguiente manera. En la Sección 2.4.1 se presentan los primeros problemas de programación entera-mixta estocásticos estudiados en la literatura y su resolución. Luego, se presenta una formulación general basada en árboles de escenarios para representar problemas estocásticos multi-etapa que será de utilidad para generar inecuaciones válidas en el Capítulo 4. En la Sección 2.4.2 se presenta literatura para Problemas de Dimensionado de Lotes Estocásticos, y se presenta una posible clasificación de la misma basada en la cantidad de etapas del problema. Finalmente, en la Sección 2.4.3, se presentan estudios específicos para Problemas de Dimensionado de Lotes con Remanufactura Estocásticos.

2.4.1. Problemas estocásticos generales

La programación entera-mixta estocástica requiere la integración de técnicas de programación entera-mixta y programación estocástica lineal. Los primeros trabajos que incorporan incertidumbre en sus parámetros, fueron en general modelos de dos etapas, en los que las decisiones de primer etapa son entera-mixtas, y las de segunda etapa se obtienen de modelos programación lineal (Sen, 2005). A continuación se presentan los primeros modelos de problemas estocásticos generales, aplicables a una diversa cantidad de situaciones de la realidad. Luego, se presenta una formulación general basada en árboles de escenarios para representar problemas estocásticos multi-etapa.

2.4.1.1. Primeros modelos

Dentro de los primeros modelos de programación estocástica con dos etapas, se destacan los trabajos de Beale (1955) y Dantzig (1955). Beale (1955) propone un enfoque de programación cuadrática para resolver casos particulares de programación estocástica con recurso simple. Dantzig (1955) formula el problema general de programación lineal con incertidumbre en sus parámetros. Tintner (1955) y van de Panne y Popp (1963) estudian las primeras aplicaciones prácticas de la programación estocástica lineal en el ámbito de la agricultura y

la ganadería.

En cuanto a estrategias de resolución, uno de los posibles enfoques a utilizar son las técnicas de descomposición, las que en general se aplican a problemas de programación matemática de gran tamaño. En el caso de problemas estocásticos, dado que la cantidad de escenarios diferentes que se deben considerar es elevada, puede ser importante tener enfoques que permitan escalar el problema. Los primeros métodos de descomposición fueron los de generación de columnas de Dantzig y Wolfe (1960) y métodos basados en el agregado de planos de cortes Benders (1962), y el método de resolución L de Van Slyke y Wets (1969). Ejemplos de resolución a través de métodos de descomposición para problemas multi-etapa se pueden encontrar en Birge (1985) y Gassmann (1990).

Una manera de representar problemas estocásticos multi-etapa, es utilizar árboles de escenarios. A continuación se presenta una formulación genérica basada en árboles de escenarios, que será de utilidad para comprender las formulaciones basadas en el agregado de inecuaciones de árbol del Capítulo 4. La formulación permite modelar apropiadamente cualquier problema estocástico multi-etapa sobre un árbol de escenarios (Römisch y Schultz, 2001).

2.4.1.2. Formulación genérica basada en árboles de escenarios

En Guan et al. (2006b) se presenta una formulación genérica para problemas de programación estocástica multi-etapa. Los parámetros siguen un proceso estocástico de tiempo discreto con T períodos. Esta estructura de información se puede representar como un árbol de escenarios $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, con T niveles, donde un nodo $i \in \mathcal{V}$, representa el estado del sistema con la información que se tiene hasta el nivel t, \dots, T . Dados los parámetros (α, β, G, A, b) , cuya estructura es la siguiente: A_{ij} y G_{ij} son matrices, α_i, β_i, b_i son vectores de la dimensión apropiada donde $i, j \in \mathcal{V}$. Se asume, sin pérdida de generalidad, que las variables \hat{x}_i , que representan las decisiones de cantidades continuas, tienen la misma dimensión entre todos los nodos. Análogamente, las variables \hat{y}_i , que representan las decisiones de cantidades enteras, también tienen la misma dimensión entre sí. El objetivo es minimizar el costo esperado.

El conjunto de nodos en el camino desde la raíz hasta el nodo i se denota $\mathcal{P}(i)$. La probabilidad de ocurrencia del estado del sistema en un nodo i se representa con p_i . La decisión (\hat{x}_i, \hat{y}_i) que se toma en el nodo i , se realiza luego de conocerse las realizaciones de los parámetros aleatorios en los nodos del camino desde la raíz hasta el nodo $(\alpha_i, \beta_i, \{G_{ij}\}_{j \in \mathcal{P}(i)}, \{A_{ij}\}_{j \in \mathcal{P}(i)}, b_i)$. Además, la decisión es no anticipativa con respecto a

futuras realizaciones.

El problema se puede modelar a través de la formulación (2.39)-(2.41):

$$\min \sum_{i \in \mathcal{V}} p_i (\alpha_i \hat{x}_i + \beta_i \hat{y}_i) \tag{2.39}$$

s.a :

$$\sum_{j \in \mathcal{P}(i)} (G_{ij} \hat{x}_j + A_{ij} \hat{y}_j) \geq b_i, \quad i \in \mathcal{V} \tag{2.40}$$

$$\hat{x}_i \in \mathbb{R}_+^p, \hat{y}_i \in \mathbb{Z}_+^n, \quad i \in \mathcal{V} \tag{2.41}$$

Esta formulación genérica para problemas estocásticos será de utilidad Capítulo 4, donde se generan inecuaciones válidas para uno de los problemas del caso de estudio. En la siguiente Sección se presenta literatura para Problemas de Dimensionado de Lotes Estocásticos, clasificándola en base a la cantidad de etapas del problema estocástico.

2.4.2. Dimensionado de lotes estocástico

Como fue mencionado al comienzo de la Sección 2.4, el proceso de decisión se lleva a cabo en determinadas etapas. Para el caso específico de Problemas de Dimensionado de Lotes, generalmente las etapas están dadas por períodos de tiempo pertenecientes al horizonte de planificación. Una posible clasificación para los Problemas de Dimensionado de Lotes Estocástico es basada en la cantidad de etapas que considera el modelo:

- **Modelos de una etapa.** En los modelos con incertidumbre de una única etapa se busca crear planes de producción para todo el horizonte de planificación. Si bien los parámetros pueden seguir una distribución de probabilidad, todas las decisiones se toman en el primer período.
- **Modelos de dos etapas.** En los problemas de dimensionado de lotes de dos etapas, se elige un período determinado, a partir del cual el valor de los parámetros no se conoce con certidumbre. De esta manera, los parámetros pertenecientes a períodos siguientes al período seleccionado son estocásticos, y siguen una distribución de probabilidad determinada.
- **Modelos multi-etapa.** Los modelos multi-etapa permiten posponer las decisiones a períodos siguientes a medida que parte de la incertidumbre es resuelta. Si bien los problemas de dos etapas pueden ser instancias particulares de esta categoría de

problemas, en la revisión de literatura a continuación nos referimos a modelos multi-etapa a los que admiten instancias con cualquier cantidad de etapas.

A continuación, se presentan algunos de los principales estudios para el Problema de Dimensionado de Lotes Estocástico. Dado que nuestro trabajo se centra en la resolución de problemas multi-etapa, se comienza detallando la literatura relacionada a problemas multi-etapa, tanto lineales como entero-mixtos. Luego, se presenta literatura para problemas con una y dos etapas.

Uno de los primeros problemas de producción multi-etapa fue presentado por Beale et al. (1980). Los autores desarrollan un método basado en recursión que permite obtener soluciones de similar calidad a las obtenidas a través de métodos de programación dinámica. Haugen et al. (2001) estudian una versión estocástica del DLSP con la posibilidad de satisfacer demanda con atraso. El problema es formulado con la técnica de escenarios separados descrita anteriormente, y luego resuelto utilizando una heurística adaptada del algoritmo de Rockafellar y Wets (1991). Este algoritmo es un método general de resolución de problemas de programación estocástica.

Ahmed et al. (2003) estudian el problema de expansión de capacidad multi-período bajo incertidumbre. Utilizando la técnica de programación entera-mixta estocástica, establecen una formulación basada en árboles de escenarios para el DLSP con incertidumbre en los parámetros. De nuestro conocimiento, este es el primer estudio en formular un problema de dimensionado de lotes estocástico utilizando árboles de escenarios. Los autores luego reformulan el problema como un problema de localización de instalaciones. La reformulación de localización de instalaciones para el DLSP de Bilde y Krarup (1977) tiene una relajación a programación lineal cuya solución óptima coincide con la del problema original (Nemhauser y Wolsey, 1988). Ahmed et al. (2003) demuestran que esto no se cumple si se adapta la formulación al problema con incertidumbre. El motivo detrás de este resultado, es que no se cumple la condición de Wagner-Within de inventario cero vista anteriormente. Sin embargo, los autores destacan que si bien las soluciones obtenidas no son necesariamente enteras, de todas maneras son útiles para disminuir el gap de dualidad.

Guan et al. (2006b) presentan una formulación genérica para problemas de programación estocástica multi-etapa. Los autores brindan un método general para generar planos de cortes a partir de la combinación de inecuaciones válidas para los caminos, haciendo uso del concepto de emparejamiento de inecuaciones (Guan et al., 2007). El método es luego puesto en práctica en la generación de inecuaciones válidas, conocidas como inecuaciones

de árbol, para las versiones estocásticas del problema del mochilero y para el problema de dimensionado de lotes estocástico. En un trabajo siguiente, Guan et al. (2006c) demuestran que las inecuaciones (l, S) presentadas anteriormente son válidas para las variantes que incorporan incertidumbre en los parámetros, al ser incluídas como inecuaciones de camino. Las inecuaciones son luego extendidas a una clase general, que combina las inecuaciones de caminos para generar inecuaciones de árbol. Esta combinación se lleva a cabo siguiendo los lineamientos descritos en Guan et al. (2006b). Los autores demuestran que esta nueva clase de inecuaciones describen caras del poliedro del conjunto de soluciones factibles. Finalmente, desarrollan una heurística de separación para las nuevas inecuaciones, de modo de seleccionar un subconjunto de estas a incorporar en el procesamiento de cada nodo en un algoritmo branch-and-bound.

Siguiendo la idea de Bilde y Krarup (1977), Guan et al. (2006a) desarrollan una formulación de localización de instalaciones para el problema de dimensionado de lotes estocástico. En relación a la incorporación de inecuaciones (l, S) , señalan que si bien su incorporación es suficiente para describir el casco convexo de las soluciones factibles para la variante determinista, ésto no se cumple para la variante estocástica incluso para problemas con sólo dos períodos. Sin embargo, la incorporación de la nueva familia de inecuaciones propuesta en Guan et al. (2006c) es suficiente para describir el casco convexo del problema de dimensionado de lotes estocástico de dos períodos. Finalmente, demuestran que las reformulaciones con inecuaciones de tipo (l, S) y las de localización de instalaciones son idénticas en términos de las cotas inferiores que proveen sus relajaciones a programación lineal.

Guan y Miller (2008) indican dos propiedades que debe cumplir al menos una de las soluciones óptimas en los problemas de dimensionado de lotes multi-etapa:

- para un nodo i cualquiera donde se decide producir, la cantidad producida más el inventario entrante, proveniente del nodo predecesor, debe ser capaz de cubrir exactamente la demanda acumulada desde i hasta algún nodo descendiente de i .
- el inventario entrante a un nodo i debe ser, para algún nodo j descendiente de i , igual a la resta de la demanda acumulada desde la raíz a j menos la demanda acumulada desde la raíz a i .

Siguiendo estas propiedades, los autores desarrollan un algoritmo de tiempo polinomial en la cantidad de nodos del árbol, basado en programación dinámica para su resolución. En Zhao y Guan (2014) se presentan formulaciones extendidas para el problema de dimensionado de lotes estocástico multi-etapa, con parámetros deterministas, basándose en las propiedades de

optimalidad estudiadas en Guan y Miller (2008).

En cuanto a estrategias de resolución heurística computacionales para problemas multi-etapa, Tang et al. (2012) y Tempelmeier y Derstroff (1996) presentan algoritmos de resolución basados en la relajación langrangeana para variantes del problema. Beraldi et al. (2006) utiliza la heurística relax-and-fix para resolver un problema en el que se debe asignar la producción de varios ítems en una cierta cantidad de máquinas. La estrategia de resolución se basa en el trabajo de Alonso et al. (2000), quienes desarrollan una heurística relax-and-fix para un problema de manejo de tráfico aéreo. Ambos trabajos aprovechan la estructura de la formulación para resolver una secuencia de subproblemas asociados a los nodos del árbol. Como se verá en el Capítulo 5, esta estrategia presenta dificultades nuevas para ser aplicada en nuestro caso de estudio.

En relación a dimensionado de lotes con dos etapas, se destacan los siguientes estudios. Zhou y Guan (2010) consideran un problema de dimensionado de lotes estocástico de dos etapas, multi-período con demanda determinista, y costos de Wagner-Within con incertidumbre. Al igual que en otros problemas de dimensionado de lotes de dos etapas, se elige un período determinado, en el cual las predicciones de los parámetros necesita ser ajustada. En este trabajo, los costos pertenecientes a períodos siguientes al período son estocásticos, y siguen una distribución de probabilidad discreta. Extendiendo el trabajo de Guan et al. (2006a) que encontraba una formulación cuya relajación a programación lineal devuelve soluciones enteras para problemas con dos períodos y dos etapas, Zhou y Guan (2010) brindan una reformulación para el problema multi-período, con dos etapas y costos de Wagner-Within que brinda soluciones enteras. Asimismo, proveen el casco convexo del problema proyectando la reformulación sobre el espacio de soluciones original.

Finalmente, en relación a problemas de una única etapa, Vargas (2009) estudia un modelo de una etapa para el DLSP con demanda estocástica y no-estacionaria, y resuelve el problema para el caso particular en el que la demanda sigue una distribución de probabilidad normal.

2.4.3. Literatura de dimensionado de lotes estocástico para problemas con remanufactura

De nuestro conocimiento, hasta el momento no existen estudios sobre versiones estocásticas de ELSRs y ELSRj utilizando técnicas de programación entera-mixta. En relación a problemas similares al caso de estudio, pero formulados con estrategias diferentes a programación matemática, se puede señalar el trabajo de Naeem et al. (2013). Los autores estudiaron un problema con Remanufactura Estocástico, modelando el problema con programación dinámica. Instancias de tamaño pequeño son resueltas a través de fórmulas matemáticas recursivas.

Macedo et al. (2016) estudian un modelo de remanufacturación estocástico particular de dos etapas, en el que las variables de producción de artículos nuevos, de remanufacturación y las variables de activación son de primera etapa, mientras que las variables de inventario, y de disposición de productos son variables de segunda etapa. Hilger et al. (2016) analizan un modelo no lineal para problema de remanufacturación multi-producto. El modelo es aproximado utilizando muestreo de promedios y un esquema de aproximación lineal por tramos.

Vu et al. (2017) estudian sistemas de desmontaje con pérdida de ventas en los que la única opción es remanufacturar, ya que no se permite producir artículos nuevos. La pérdida de ventas corresponde a los casos en los que no es obligatorio satisfacer toda la demanda, y se acepta pagar una penalidad por cada unidad de demanda no vendida. El desmontaje consiste de tres procesos: en primer lugar los retornos se desmontan obteniéndose un conjunto de partes recuperables, en segundo lugar esas partes se reacondicionan, y finalmente, las partes listas se reensamblan, obteniéndose los artículos con los que se satisface la demanda. Para su resolución, los autores investigan la incorporación de inecuaciones válidas del problema de dimensionado de lotes con pérdida de ventas de Loparic et al. (2001). Este trabajo es complementado por Quezada et al. (2020), quienes estudian la incorporación de inecuaciones válidas de árbol, generadas a partir de las inecuaciones de camino mencionadas anteriormente, e incorporadas en un mecanismo heurístico para la resolución del problema. Junior y Filho (2017) establecen un modelo de programación dinámica para resolver un problema real de desmontaje, para la remanufacturación de embragues para vehículos.

Fang et al. (2017) presentan un modelo con demandas estocásticas para los productos manufacturados y remanufacturados, y planificación de capacidad de recursos. El problema

es formulado mediante árboles de escenarios, y resuelto con una heurística basada en la relajación lagrangiana. Kilic et al. (2018) proponen políticas heurísticas para un problema con remanufactura estocástico con restricciones de nivel de servicio.

Capítulo 3

Dimensionado de Lotes con Remanufactura Estocástico

Con la motivación de responder a la creciente problemática ambiental y disminuir el desperdicio de recursos naturales, existe un interés por cambiar los esquemas de producción tradicionales a un esquema de reutilización de materiales (Suzanne et al., 2020). El concepto de economía circular definido en los trabajos de Homrich et al. (2018) y Reike et al. (2018), es utilizado en un estudio reciente de Suzanne et al. (2020) en el que se presenta una división en temáticas relacionadas con este concepto: manejo del desperdicio, reciclaje, sustentabilidad, entre otros. Dentro de la temática de reciclaje de productos y materiales, se pueden encontrar las operaciones y procesos de remanufactura y reacondicionamiento. Mientras que en el reacondicionamiento, los productos se limpian y reparan para ser vendidos bajo una garantía especial, en la remanufactura se pueden realizar diversas operaciones como reparación, desmontaje y reemplazo de partes con el objetivo de garantizar productos con igual calidad que productos nuevos. Como fue desarrollado en el Capítulo 2, la temática de la remanufactura es en sí misma un área que alberga distintos tipos de trabajos, dependiendo de la etapa del proceso que se desea estudiar. En este trabajo, se hace foco en una extensión que ha tomado particular interés en los últimos años, como son los problemas de Dimensionado de Lotes con Remanufactura. En este capítulo se desarrollan modelos estocásticos para problemas de remanufactura. Partiendo de la literatura referente a los problemas con remanufactura discutida en el capítulo anterior, se aborda el modelado mediante programación entera-mixta estocástica.

El desarrollo del capítulo es el siguiente. En la Sección 3.1 se describen variantes estocásticas de problemas de dimensionado de lotes con remanufactura, que llamaremos Stochastic Economic Lot-Sizing with Remanufacturing and separate setups (SELSRs) y Stochastic

Economic Lot-Sizing with Remanufacturing and joint setups (SELSRj). En la Sección 3.2 se presentan las entidades necesarias para representar el problema mediante árboles de escenarios. Finalmente, en la Sección 3.3 se presenta las primeras formulaciones para las variantes mencionadas.

3.1. Descripción del problema

En este trabajo, se consideran variantes estocásticas de los problemas ELSRs y ELSRj, que llamaremos SELSRs y SELSRj, con incertidumbre en los valores de demanda y retornos. La denominación ELSRs y ELSRj ha sido utilizada por distintos autores para referirse a problemas de remanufactura similares, con algunas diferencias en los supuestos a partir de los que se formula el modelo. Versiones deterministas que sirven de base a este trabajo se investigaron recientemente en Retel Helmrich et al. (2014), Cunha y Melo (2016) y Ali (2017). En este trabajo, estudiaremos variantes del problema en las que permite que la demanda se pueda satisfacer mediante tanto el proceso de producción de artículos nuevos como el proceso de remanufactura. Así, el proceso de remanufactura transforma artículos usados que son devueltos al origen, y llamaremos retornos, en artículos listos que se pueden utilizar para satisfacer la demanda. Dado que suponemos que los productos remanufacturados son tan buenos como los productos nuevos, la demanda se puede satisfacer indistintamente por cualquiera de los dos. Además, se incorporan costos de producción unitarios a la formulación, y tanto los costos de configuración como los de inventario son dinámicos. Una representación del sistema de inventarios de este tipo de sistemas se puede ver en la Figura 3.1

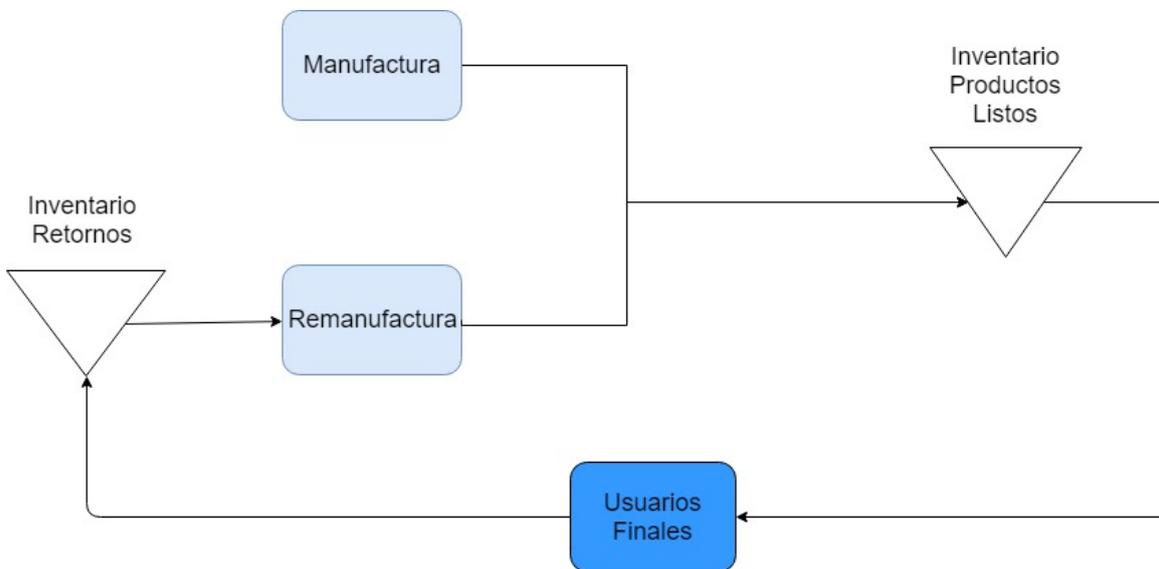


Figura 3.1 Esquema de inventarios para SELSRs y SELSRj

La incorporación de parámetros estocásticos es un aspecto que toma especial interés en problemas con remanufactura. Retel Helmrich et al. (2014) destacan que si bien existen ciertos aspectos de la realidad que se pueden capturar con un modelo determinista, la suposición

de demanda y retornos deterministas puede ser demasiado restrictiva en algunos contextos. Ali (2017) señala que las actividades asociadas con planificación de producción son aún más complejas al considerar remanufactura, ya que incorporan una mayor incertidumbre en sus parámetros. El autor considera que existen buenas oportunidades de investigar problemas que agreguen incertidumbre a la demanda y cantidad de retornos.

3.2. Representación mediante árboles de escenarios

Como fue mencionado en la sección anterior, trabajaremos sobre problemas con remanufactura en los que asumimos que los parámetros de demanda y retornos tienen incertidumbre y siguen un proceso estocástico de tiempo discreto con espacio de probabilidad finito. La estructura de información resultante es la de un árbol de escenarios. Para mantener la mayor generalidad posible, se considera que el árbol de escenarios es completo. De esta manera, siempre es posible contemplar la existencia de incertidumbre al pasar de un período al siguiente.

Se considera un árbol con raíz (arborescencia) compuesto por un conjunto de nodos N . Cada nodo $n \in N$ correspondiente a un período t , nivel en el árbol, constituye el estado de la realidad hasta dicho período, con la información con la que se cuenta hasta el momento. Dado que se trabaja con árboles completos, el número de nodos del conjunto N es exponencial en la cantidad de periodos del conjunto T . Si la aridad del árbol es g , la cantidad de nodos del árbol está dada por la siguiente relación $|N| = g^{T-1} + 1$. Existe un único nodo raíz al que se le asigna el índice $n = 1$. Las variables de decisión asociadas a este nodo representan las decisiones del primer período, las cuales son deterministas, ya que no dependen de la realización de ningún evento aleatorio. Cada nodo i que no es raíz del árbol, tiene un nodo antecesor en el árbol, denotado $a(i)$. A su vez, cada nodo i del árbol es nodo raíz de un subárbol, que contiene al nodo i y a todos sus descendientes. Cada nodo tiene una probabilidad de ocurrencia π_n , de modo que la suma de las probabilidades de los nodos asociados a un período debe valer uno. Finalmente, se incorporan parámetros para representar los caminos entre dos nodos, y se denomina nodo hoja a los nodos asociados al último período del horizonte de planificación. Un ejemplo de árbol de escenarios 2-ario con cuatro períodos, junto a las entidades mencionadas se puede ver en la Figura 3.2. A continuación se presenta un listado con la notación de las estructuras de datos necesarias para representar árboles de escenarios:

- $a(n)$: nodo antecesor del nodo n en el árbol
- π_n : probabilidad de ocurrencia del nodo n
- $\mathcal{P}(m, n)$: nodos en el camino del árbol desde el nodo m al n . El camino del nodo raíz al nodo n se denota como $\mathcal{P}(n)$
- $\mathcal{L}(n)$: nodos hojas del subárbol con nodo raíz n

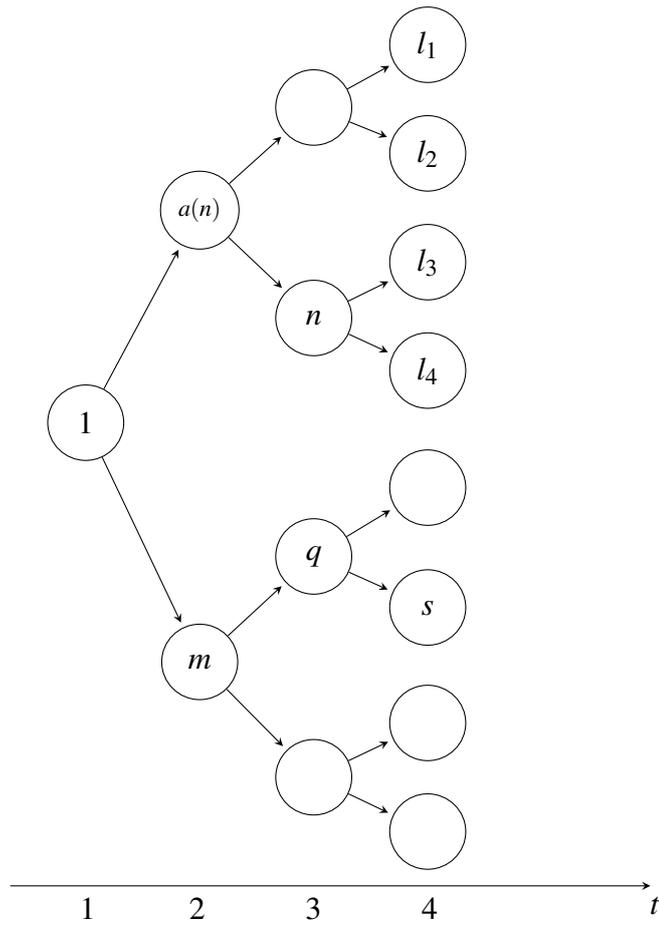


Figura 3.2 Árbol 2-ario de cuatro períodos. En la figura se observan los períodos al que pertenece cada nodo y el antecesor de un nodo n . En este ejemplo, además se tiene: $\mathcal{L}(n) = \{l_3, l_4\}$, $\mathcal{L}(a(n)) = \{l_1, \dots, l_4\}$, $\mathcal{P}(n) = \{1, a(n), n\}$, $\mathcal{P}(m, s) = \{m, q, s\}$.

3.3. Formulación natural

Habiéndose definido las entidades que serán necesarias para formular los problemas con árboles de escenarios, en esta sección se definen parámetros y variables para los problemas SELSRs y SELSRj descritos en la Sección 3.1. Un resumen con la notación utilizada en estas definiciones se puede ver en la Tabla 3.1. Luego, se presentan formulaciones para ambos problemas, a las que nos referiremos como formulaciones naturales en el resto del presente trabajo.

Parámetros

Para cada nodo $n \in N$ se tiene una demanda d_n a satisfacer y una determinada cantidad de retornos r_n . A su vez, en cada nodo se considera los siguientes costos:

- h_n^s, h_n^r : costos de almacenar productos listos y de almacenar retornos respectivamente.
- K_n^m, K_n^r, K_n : para el problema con líneas de producción separadas, K_n^m y K_n^r son los costos de configuración para la producción de artículos nuevos y para la remanufacturación respectivamente. En caso de tratarse del problema con una única línea de producción, se tiene un único costo K_n para ambas actividades.
- p_n^m, p_n^r : costos de producción unitarios para la producción de artículos nuevos y para la remanufacturación respectivamente.

Variables

Se tienen dos familias de variables, x_n^m y x_n^r para representar la cantidad de productos nuevos y remanufacturados en un nodo n , respectivamente. Hay dos esquemas para modelar las variables de activación, dependiendo si se tiene una línea de producción compartida, o líneas de producción dedicadas. Para el caso con líneas de producción separadas, se tiene variables binarias para activar la producción de artículos nuevos y_n^m y para la remanufactura y_n^r . Para el caso con línea de producción única, solamente se tiene la familia de variables y_n . Finalmente, se tiene variables para los inventarios de artículos listos I_n^s y para los artículos retornados en cada nodo I_n^r .

Tabla 3.1 Notación de parámetros y variables para los problemas SELSRs y SELSRj

N	conjunto de nodos con subíndice n
d_n	demanda en el nodo n
r_n	retorno en el nodo n
h_n^s	costos unitarios de almacenar en inventario en el nodo n para artículos nuevos
h_n^r	costos unitarios de almacenar en inventario en el nodo n para artículos remanufacturados
K_n^m , K_n^r	costo de activar producción en el nodo n para artículos nuevos y remanufacturados respectivamente
K_n	costo de activar producción en el nodo n para el problema con costo de configuración común
p_n^m	costo de producción de artículos nuevos unitario en el nodo n
p_n^r	costo de remanufactura unitario en el nodo n
x_n^m	productos nuevos producidos en el nodo n
x_n^r	productos remanufacturados en el nodo n
y_n^m	1 si se activa la producción de artículos nuevos en el nodo n , 0 en caso contrario
y_n^r	1 si se activa la remanufactura en el nodo n , 0 en caso contrario
y_n	1 si se activa la producción de artículos nuevos o remanufactura en el nodo n para el problema con costo de configuración común, 0 en caso contrario
I_n^s	inventario de productos listos en n
I_n^r	inventario de retornos en n

3.3.1. Modelo con configuración en líneas de producción separadas (costos de setup separados)

Partiendo de la definición de conjuntos, parámetros y variables anterior, en esta sección se presenta una formulación, que se denominará Natural, para el problema con líneas de producción separadas, SELSRs.

$$(\text{SELSRs}) : \min \sum_{n \in N} \pi_n (K_n^m y_n^m + K_n^r y_n^r + h_n^s I_n^s + h_n^r I_n^r + p_n^m x_n^m + p_n^r x_n^r) \quad (3.1)$$

s.a :

$$I_{a(n)}^s + x_n^m + x_n^r = I_n^s + d_n, \quad n \in N \quad (3.2)$$

$$I_{a(n)}^r + r_n = I_n^r + x_n^r, \quad n \in N \quad (3.3)$$

$$I_{a(1)}^s = I_{a(1)}^r = 0. \quad (3.4)$$

$$x_n^m \leq M_n^m y_n^m, \quad n \in N \quad (3.5)$$

$$x_n^r \leq M_n^r y_n^r, \quad n \in N \quad (3.6)$$

$$x_n^m, x_n^r, I_n^s, I_n^r \geq 0, \quad n \in N \quad (3.7)$$

$$y_n^m, y_n^r \in \{0, 1\}. \quad n \in N \quad (3.8)$$

donde $M_n^m = \max_{\lambda \in \mathcal{L}(n)} \left\{ \sum_{\mu \in \mathcal{P}(n, \lambda)} d_\mu \right\}$

y donde $M_n^r = \min \left\{ \sum_{\mu \in \mathcal{P}(n)} r_\mu, \max_{\lambda \in \mathcal{L}(n)} \left\{ \sum_{\mu \in \mathcal{P}(n, \lambda)} d_\mu \right\} \right\}$.

El objetivo (3.1) es minimizar el valor esperado de los costos asociados a los procesos de producir artículos nuevos y remanufacturar en todo el árbol de escenarios. Las restricciones (3.2) indican que la demanda se puede satisfacer con productos nuevos, con productos remanufacturados ó con productos que estaban presentes en el inventario de productos listos al comenzar el período. El excedente pasa del nodo antecesor al siguiente como inventario de productos listos. (3.3) son restricciones de balance de inventario para los retornos, e indican que la remanufacturación se puede llevar a cabo con retornos de un nodo o presentes en el inventario del nodo antecesor. El excedente pasa del nodo antecesor al siguiente como inventario de retornos. En las restricciones (3.4) se indica que en el nodo raíz tanto los inventarios disponibles de productos listos como de retornos están vacíos. (3.5) son restricciones de activación de la producción para la producción de artículos nuevos, e indican que para poder producir artículos nuevos en un nodo se debe activar su variable asociada y_n^m . (3.6) son restricciones de activación de la producción para la remanufacturación, e indican que para

poder remanufacturar en un nodo se debe activar su variable asociada y_n^r .

Como fue mencionado en el Capítulo 2, Pochet y Wolsey (2006) indican que las restricciones de “M grande” en los problemas de dimensionado de lotes pueden llevar a malas relajaciones de programación lineal. Por lo tanto, para la selección de M_n^m y M_n^r en las restricciones (3.5) y (3.6) puede ser conveniente elegir como valores los menores posibles, con el objetivo de intentar ajustar esas restricciones. De esta manera, la formulación será más cercana al casco convexo del problema. El valor de M_n^m se puede seleccionar teniendo en cuenta que en un nodo n no se necesita producir más artículos nuevos que el máximo de las demandas acumuladas entre todos los posibles caminos desde n hasta sus hojas. El valor de M_n^r se elige teniendo en cuenta dos propiedades. En primer lugar, en un nodo n no se necesitará remanufacturar más productos que el máximo de las demandas acumuladas entre todos los posibles caminos desde n hasta las hojas. En segundo lugar, en un nodo n no será posible remanufacturar más productos que la cantidad de retornos acumulados desde el nodo raíz hasta n . Por lo tanto, la cantidad de productos que se remanufacturan en n será el mínimo entre esas dos cantidades.

3.3.2. Modelo con configuración en una línea de producción común (costo de setup común)

En esta sección se presenta una primer formulación utilizando la estructura de árboles de escenarios detalladas al principio de este capítulo, para el problema con una única línea de producción, SELSRj.

$$(\text{SELSRj}) : \min \sum_{n \in N} \pi_n (K_n y_n + h_n^s I_n^s + h_n^r I_n^r + p_n^m x_n^m + p_n^r x_n^r) \quad (3.9)$$

s.a :

$$I_{a(n)}^s + x_n^m + x_n^r = I_n^s + d_n, \quad n \in N \quad (3.10)$$

$$I_{a(n)}^r + r_n = I_n^r + x_n^r, \quad n \in N \quad (3.11)$$

$$I_{a(1)}^s = I_{a(1)}^r = 0 \quad (3.12)$$

$$x_n^m + x_n^r \leq M_n y_n, \quad n \in N \quad (3.13)$$

$$x_n^m, x_n^r, I_n^s, I_n^r \geq 0, \quad n \in N \quad (3.14)$$

$$y_n \in \{0, 1\}. \quad n \in N \quad (3.15)$$

$$\text{donde } M_n = \max_{\lambda \in \mathcal{L}(n)} \left\{ \sum_{\mu \in \mathcal{P}(n, \lambda)} d_\mu \right\}$$

La interpretación de las restricciones es análoga a la presentada para el modelo SELSRs, salvo para las restricciones de activación de la producción, y la selección del parámetro M_n . El objetivo (3.9) es minimizar el valor esperado de los costos asociados a los procesos de producir artículos nuevos y remanufacturar en todo el árbol de escenarios. Las restricciones (3.10) indican que la demanda se puede satisfacer con productos nuevos, con productos remanufacturados ó con productos que estaban presentes en el inventario de productos listos al comenzar el período. El excedente pasa del nodo antecesor al siguiente como inventario de productos listos. (3.11) son restricciones de balance de inventario para los retornos, e indican que la remanufacturación se puede llevar a cabo con retornos de un nodo o presentes en el inventario del nodo antecesor. El excedente pasa del nodo antecesor al siguiente como inventario de retornos. En las restricciones (3.12) se indica que en el nodo raíz tanto los inventarios disponibles de productos listos como de retornos están vacíos. (3.13) son restricciones de activación de la producción tanto de productos nuevos como de productos remanufacturados, e indican que para poder producir en un nodo se debe activar su variable asociada y_n .

Siguiendo el objetivo planteado para el SELSRs de tener una formulación lo más cercana al casco convexo del problema posible, se analiza la selección del parámetro M_n en las restricciones (3.13). El valor de M_n se puede seleccionar teniendo en cuenta que en un nodo n no se necesita producir más productos que el máximo de las demandas acumuladas entre todos los posibles caminos desde n hasta sus hojas.

3.3.3. Complejidad

Los problemas SELSRs y SELSRj pertenecen a la clase NP-hard. Esto se debe a que, como fue visto en el Capítulo 2, sus versiones deterministas son NP-hard. Dado que el problema determinista es un caso particular del que incorpora incertidumbre, en el que el árbol de escenarios tiene aridad uno, el problema estocástico también es NP-hard. Cabe mencionar que independientemente de la extensión del problema al considerar remanufactura, la incorporación de incertidumbre en los problemas representa una dificultad en si misma. Si bien existen algoritmos para la versión estocástica del DLSP que corren en tiempo pseudo-polinomial en la cantidad de nodos del árbol de escenarios, esta cantidad crece exponencialmente en la cantidad de períodos que se considera para el horizonte de planificación. Por estos motivos, es de esperar que incluso instancias del problema con horizontes de planificación con pocos períodos sean difíciles de resolver a nivel práctico.

Capítulo 4

Reformulaciones e Inecuaciones Válidas

Con la intención de obtener formulaciones más próximas al casco convexo que la formulación natural, en este capítulo se presentan distintas reformulaciones para las variantes SELSRs y SELSRj. En la Sección 4.1 se presentan formulaciones extendidas basadas en la reformulación propuesta originalmente por Bilde y Krarup (1977) para el DLSP. En este contexto se presentan reformulaciones de Localización de Instalaciones en las que se desagregan las variables de producción de artículos nuevos y de remanufactura, con el objetivo de especificar más detalladamente el esquema de producción. En la Sección 4.2 se deducen inecuaciones válidas a partir de las inecuaciones (I, S) de Barany et al. (1984b) y Barany et al. (1984a) para los problemas SELSRs y SELSRj. En la Sección 4.3 se describe el emparejamiento de inecuaciones. A partir de este procedimiento, se desarrolla una estrategia para combinar las inecuaciones válidas de la Sección 4.2 para el SELSRs. Como resultado, se obtienen nuevas inecuaciones, denominadas de árbol. La incorporación de inecuaciones tanto en las formulaciones de la Sección 4.2, como en la Sección 4.3, permite mejorar las cotas obtenidas mediante la relajación a programación lineal de las formulaciones, y potencialmente puede mejorar el desempeño de las estrategias de resolución de ramificado y acotamiento (branch-and-bound).

4.1. Reformulación localización de instalaciones

De modo similar a lo que sucede en el problema clásico, en las variantes con remanufactura se tienen restricciones de activación de producción, en las que el tamaño del lote en un período se encuentra acotado o bien por la demanda acumulada hasta el último período, o bien por la cantidad de retornos que se han recibido hasta el momento. Esta cota puede ser holgada, especialmente para los primeros períodos, donde es de esperarse que no haya necesidad de producir cantidades tan elevadas. En busca de ajustar dicha cota, en las reformulaciones de Localización de Instalaciones de Bilde y Krarup (1977) se desagregan las variables de producción definiendo nuevas variables para estas cantidades. Estrategias similares se han aplicado a problemas deterministas (Cunha y Melo, 2016; Retel Helmrich et al., 2014; Ali, 2017). De manera similar, en este trabajo se presentan reformulaciones de Localización de Instalaciones en las que se descomponen las variables de manufacturación y remanufacturación. Dado un nodo i y un nodo j , de modo que existe un camino desde el nodo i hasta el nodo j , se extiende la formulación con el agregado de las siguientes variables:

- w_{ij}^{sr} : cantidad de productos remanufacturados en el nodo i para satisfacer la demanda en el nodo j .
- w_{ij}^{sm} : cantidad de productos nuevos producidos en el nodo i para satisfacer la demanda en el nodo j .
- w_{ij}^r : cantidad de productos remanufacturados en el nodo j con productos que fueron retornados en el nodo i .

Para el caso del problema con remanufactura estocástico, procuramos mejorar las cotas representadas en los parámetros M_n^1 , M_n^2 para SELSRs, y M_n para SELSRj. Las nuevas redefiniciones de las variables de lote permiten especificar más detalladamente el esquema de producción óptimo indicando no sólo dónde se produce sino también cuándo será consumida dicha producción, ó indicando de cual período provienen los retornos con los que se está remanufacturando. De esta manera, como se verá en el estudio computacional del Capítulo 6, se logra ajustar las restricciones de activación originales. A continuación se presentan las reformulaciones para los problemas SELSRs y SELSRj, que llamaremos FLSELSRs y FLSELSRj respectivamente.

4.1.1. Localización de instalaciones setup separado

Luego el agregado de las variables de producción desagregadas, el modelo Localización de Instalaciones para la variante con líneas de producción separadas es el siguiente,

$$(\text{FLSELSRs}) : \min \sum_{n \in N} \pi_n (K_n^m y_n^m + K_n^r y_n^r + h_n^s I_n^s + h_n^r I_n^r + p_n^m x_n^m + p_n^r x_n^r)$$

s.a :

$$(3.2) - (3.8)$$

$$x_n^m \geq \sum_{k \in \mathcal{P}(n,j)} w_{nk}^{sm}, \quad n \in N, j \in \mathcal{L}(n) \quad (4.1)$$

$$x_n^r = \sum_{j \in \mathcal{P}(n)} w_{jn}^r, \quad n \in N \quad (4.2)$$

$$w_{nj}^{sm} \leq d_j y_n^m, \quad j \in N, n \in \mathcal{P}(j) \quad (4.3)$$

$$w_{nj}^{sr} \leq d_j y_n^r, \quad j \in N, n \in \mathcal{P}(j) \quad (4.4)$$

$$w_{jn}^r \leq r_j y_n^r, \quad n \in N, j \in \mathcal{P}(n) \quad (4.5)$$

$$\sum_{j \in \mathcal{P}(n)} w_{jn}^{sr} + w_{jn}^{sm} = d_n, \quad n \in N \quad (4.6)$$

$$\sum_{i \in \mathcal{P}(n,j)} w_{ni}^r \leq r_n, \quad n \in N, j \in \mathcal{L}(n) \quad (4.7)$$

$$\sum_{i \in \mathcal{P}(n)} w_{in}^r \geq \sum_{k \in \mathcal{P}(n,j)} w_{nk}^{sr}, \quad n \in N, j \in \mathcal{L}(n) \quad (4.8)$$

$$w_{nj}^{sm}, w_{nj}^{sr}, w_{jn}^r \geq 0, \quad n \in N, j \in \mathcal{L}(n) \quad (4.9)$$

Las restricciones (4.1) relacionan las variables originales de manufacturación de productos nuevos con las variables que especifican el nodo destino de la producción. Para cada nodo y cada hoja de un nodo, la producción de artículos nuevos en el nodo debe ser igual o mayor a la suma de las nuevas variables de producción con origen en el nodo, y con destino nodos del camino entre el nodo y una hoja. Se puede notar que en esta familia de restricciones se cumple la igualdad específicamente en el camino de mayor producción desde el nodo, dado que de lo contrario se incurrirá en costos innecesarios de producción, o de almacenar en inventario. Las restricciones (4.2) relacionan las variables originales de remanufactura con las nuevas variables de remanufactura que especifican el nodo de donde provienen los retornos que están siendo remanufacturados. Para cada nodo, lo que se remanufactura en él debe igualar la suma de las nuevas variables de remanufactura del nodo, con retornos provenientes de nodos en el camino de la raíz hasta dicho nodo.

Como fue mencionado, las restricciones de activación de la formulación inicial manejan cotas de demanda acumulada y retornos acumulados que pueden resultar holgadas. Las nuevas variables incorporadas nos permiten saber dónde será consumida la producción de determinado nodo. Haciendo uso de esta información, se puede ajustar las restricciones de activación de producción como en las restricciones (4.3) para la manufacturación y (4.4) para la remanufacturación. De esta manera, la producción en un nodo que es utilizada para satisfacer la demanda de otro está acotada únicamente por dicha demanda cuando se decide producir. De modo similar, las restricciones (4.5) son de activación, y limitan la remanufactura de retornos provenientes de un determinado nodo j hacia otro nodo, con la cantidad de retornos recibidos en j .

Las restricciones (4.6) imponen el cumplimiento de la demanda en cada nodo. Las restricciones (4.7) aseguran que, dado un nodo y un camino de dicho nodo a una de sus hojas, la cantidad de retornos debe ser suficiente para cumplir con la decisión de remanufacturar productos desde dicho nodo a los nodos del camino. Finalmente, (4.8) asegura que, dado un nodo y una hoja de su subárbol, la cantidad total de retornos recibidos en el camino desde la raíz que son remanufacturados en el nodo, debe ser mayor o igual que la cantidad de artículos remanufacturados en el nodo para satisfacer la demanda del camino hasta la hoja.

4.1.2. Localización de instalaciones setup común

El modelo Localización de Instalaciones para la variante con una única línea de producción es el siguiente,

$$(\text{FLSELSRj}) : \min \sum_{n \in N} \pi_n (K_n y_n + h_n^s I_n^s + h_n^r I_n^r + p_n^m x_n^m + p_n^r x_n^r)$$

s.a :

$$(3.10) - (3.15)$$

$$x_n^m \geq \sum_{k \in \mathcal{P}(n,j)} w_{nk}^{sm}, \quad n \in N, j \in \mathcal{L}(n) \quad (4.10)$$

$$x_n^r = \sum_{j \in \mathcal{P}(n)} w_{jn}^r, \quad n \in N \quad (4.11)$$

$$w_{nj}^{sm} + w_{nj}^{sr} \leq d_j y_n, \quad j \in N, n \in \mathcal{P}(j) \quad (4.12)$$

$$w_{jn}^r \leq r_j y_n, \quad n \in N, j \in \mathcal{P}(n) \quad (4.13)$$

$$\sum_{j \in \mathcal{P}(n)} w_{jn}^{sr} + w_{jn}^{sm} = d_n, \quad n \in N \quad (4.14)$$

$$\sum_{i \in \mathcal{P}(n,j)} w_{ni}^r \leq r_n, \quad n \in N, j \in \mathcal{L}(n) \quad (4.15)$$

$$\sum_{i \in \mathcal{P}(n)} w_{in}^r \geq \sum_{k \in \mathcal{P}(n,j)} w_{nk}^{sr}, \quad n \in N, j \in \mathcal{L}(n) \quad (4.16)$$

$$w_{nj}^{sm}, w_{nj}^{sr}, w_{jn}^r \geq 0, \quad n \in N, j \in \mathcal{L}(n) \quad (4.17)$$

Exceptuando las restricciones de activación de remanufacturación de retornos, las restricciones de esta reformulación se pueden interpretar de manera análoga a la presentada para el modelo FLSELSRs. Las restricciones (4.10) relacionan las variables originales de manufacturación de productos nuevos con las variables que especifican el nodo destino de la producción. Para cada nodo y cada hoja de un nodo, la producción de artículos nuevos en el nodo debe ser igual o mayor a la suma de las nuevas variables de producción con origen en el nodo, y con destino nodos del camino entre el nodo y una hoja. De esta manera, y dado que las variables de producción están controladas por la función objetivo, en esta familia de restricciones se cumple la igualdad específicamente en el camino de mayor producción desde el nodo. Las restricciones (4.11) relacionan las variables originales de remanufactura con las nuevas variables de remanufactura que especifican el nodo de donde provienen los retornos que están siendo remanufacturados. Para cada nodo, lo que se remanufactura en él debe igualar la suma de las nuevas variables de remanufactura del nodo, con retornos

provenientes de nodos en el camino de la raíz hasta dicho nodo.

Las restricciones de activación de producción (4.12) se diferencian de las restricciones FLSELSRs en que la suma de las variables desagregadas de producción de artículos nuevos y remanufacturación se ven acotadas por la demanda del nodo destino, en lugar de tenerse restricciones separadas para cada proceso de producción. Las restricciones (4.13) son análogas al FLSELSRs, y limitan la remanufactura de retornos provenientes de un determinado nodo j hacia otro nodo, con la cantidad de retornos recibidos en j .

Las restricciones (4.14) imponen el cumplimiento de la demanda en cada nodo. Las restricciones (4.15) aseguran que, dado un nodo y un camino de dicho nodo a una de sus hojas, la cantidad de retornos debe ser suficiente para cumplir con la decisión de remanufacturar productos desde dicho nodo a los nodos del camino. Finalmente, (4.16) asegura que, dado un nodo y una hoja de su subárbol, la cantidad total de retornos recibidos en el camino desde la raíz que son remanufacturados en el nodo, debe ser mayor o igual que la cantidad de artículos remanufacturados en el nodo para satisfacer la demanda del camino hasta la hoja.

4.2. Reformulación mediante el agregado de inecuaciones válidas

Con el objetivo de obtener formulaciones más ajustadas al casco convexo, en esta sección se deducen inecuaciones válidas, tomando como base inecuaciones anteriormente presentadas en el Capítulo 2 para variantes deterministas. Si la incorporación de estas nuevas inecuaciones es exitosa, las relajaciones a programación lineal de las formulaciones devolverán cotas más próximas al óptimo, pudiendo mejorar el desempeño de las estrategias de resolución de ramificado y acotamiento (branch-and-bound).

Para problemas estocásticos, las familias de restricciones a agregar se pueden clasificar como inecuaciones de camino, o inecuaciones de árbol (Guan et al., 2006b). Las inecuaciones de camino son inecuaciones en las que las variables de decisión involucradas son únicamente las correspondientes a nodos de un camino desde la raíz a una hoja. Las inecuaciones de árbol, por otro lado, pueden involucrar variables correspondientes a distintos escenarios, y se pueden generar a partir de la combinación de inecuaciones de camino. En esta sección se verán inecuaciones de camino para las variantes SELSRs y SELSRj, mientras que inecuaciones de árbol se presentarán en la Sección 4.3.

Las inecuaciones utilizadas para problemas con remanufactura deterministas de la literatura en general han sido adaptaciones de las inecuaciones (I, S) de Barany et al. (1984b) y Barany et al. (1984a). Dado que la cantidad de inecuaciones (I, S) que se pueden generar es exponencial en la cantidad de períodos, puede resultar poco conveniente agregarlas todas. Esta situación puede ser un desafío aún más grande para variantes estocásticas, cuya cantidad de caminos y escenarios es en sí misma exponencial en la cantidad de períodos. Por lo tanto, en este estudio procuramos fortalecer las formulaciones trabajando con la adaptación de un subconjunto de esta familia de inecuaciones, específicamente las inecuaciones (I, S, WW) . Como fue visto en el Capítulo 2, la inclusión de inecuaciones (I, S, WW) es suficiente para obtener el casco convexo de las soluciones factibles del problema clásico DLSP cuando se consideran costos de Wagner-Within (Pochet y Wolsey, 1994).

4.2.1. Inecuaciones válidas para SELSRs

Para el SELSRs estudiamos la inclusión de cuatro familias de inecuaciones basadas en las Proposiciones 2 y 4 de la Sección 2.3.3.2. Para su planteo, debemos en primer lugar definir los parámetros de demanda acumulada y retornos acumulados para problemas estocásticos:

Definición 7 (Demanda acumulada y retornos acumulados) Sean los nodos i, j tales que existe un camino desde la raíz hasta j que contiene a i . Se define la demanda acumulada D_{ij} entre los nodos i y j , y los retornos acumulados R_{ij} entre i y j de la siguiente manera:

$$D_{ij} = \sum_{n \in \mathcal{P}(i,j)} d_n$$

$$R_{ij} = \sum_{n \in \mathcal{P}(i,j)} r_n$$

En primer lugar se presentarán dos familias de inecuaciones inspiradas en la versión determinista estudiada por Retel Helmrich et al. (2014) y presentadas en el Capítulo 2. Como fue visto, estas inecuaciones fueron luego utilizadas para el problema determinista en Cunha y Melo (2016) y en Ali (2017). En este trabajo las inecuaciones son adaptadas a la variante estocástica acorde a la siguiente proposición:

Proposición 8 Sean los nodos i, j tales que existe un camino desde la raíz hasta j que contiene a i . Las siguientes inecuaciones son válidas para el SELSRs:

$$I_{a(i)}^S + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^m + y_n^r) \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.18)$$

$$I_j^r + \sum_{n \in \mathcal{P}(i,j)} R_{in}y_n^r \geq R_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.19)$$

Prueba

La validez de la Proposición 8 se puede entender de la siguiente manera. Tomando como ejemplo las familias de la Proposición 8, se puede observar la siguiente propiedad: dados dos nodos i, j tales que existe un camino desde el origen hasta j que contiene a i , la restricción resultante coincide con una inecuación (l, S, WW) para el problema con remanufactura determinista que se obtiene considerando únicamente los nodos desde la raíz a cualquiera de los nodos del conjunto $\mathcal{L}(j)$. Es decir, las variables involucradas son únicamente las correspondientes a un escenario del árbol y además la inecuación coincide con una inecuación válida determinista aplicada sobre un único escenario del árbol, por lo que es válida para ese escenario. Dado que la inecuación es válida para dicho camino, las soluciones factibles

de este subproblema deben cumplir la restricción. Dado que las soluciones factibles del problema estocástico proyectadas sobre las variables del subproblema, son un subconjunto de las del subproblema (porque el problema original es más restrictivo), la inecuación también deberá ser válida para todo el árbol.

QEP

Se puede ajustar la inecuación (4.18) observando que para que una solución sea factible, la cantidad remanufacturada no puede superar la cantidad de retornos que se recibieron desde el nodo inicial hasta el nodo en consideración:

$$I_{a(i)}^s + \sum_{n \in \mathcal{P}(i,j)} D_{nj} y_n^m + \sum_{n \in \mathcal{P}(i,j)} \min(D_{nj}, R_{1n}) y_n^r \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.20)$$

Las inecuaciones de Retel Helmrich et al. (2014) fueron extendidas por Ali (2017), con la incorporación de dos nuevas familias de inecuaciones. Estas familias de inecuaciones pueden ser adaptadas a la variante estocástica acorde a la siguiente proposición:

Proposición 9 Sean los nodos i, j tales que existe un camino desde el origen hasta j que contiene a i . Las siguientes inecuaciones son válidas para SELSRs:

$$I_{a(i)}^s + \sum_{n \in \mathcal{P}(i,j)} D_{nj} (y_n^r) + \sum_{n \in \mathcal{P}(i,j)} (I_n^s - I_{a(n)}^s - I_{a(n)}^r - r_n + I_n^r + d_n) \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.21)$$

$$I_{a(i)}^s + \sum_{n \in \mathcal{P}(i,j)} D_{nj} (y_n^m) + \sum_{n \in \mathcal{P}(i,j)} (r_n + I_{a(n)}^r - I_n^r) \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.22)$$

Prueba

El razonamiento para la Proposición 9, es análogo al de la Proposición 8, obteniéndose inecuaciones de tipo (l, S) de Ali (2017).

QEP

Un estudio computacional preliminar permitió corroborar que la incorporación de las formulaciones de Ali (2017) permite mejorar las cotas de relajación a programación lineal, con respecto a únicamente incorporar las presentadas por Retel Helmrich et al. (2014). A continuación se presentarán versiones equivalentes de las cuatro familias de restricciones, las que serán útiles para la implementación de restricciones de árbol.

4.2.1.1. Inecuaciones equivalentes sin variables de inventario para SELSRs

Las cuatro familias de inecuaciones anteriores se pueden reformular eliminando las variables de inventario, similar a lo presentado por Barany et al. (1984a) para el problema clásico, y Ali (2017) para problemas con remanufactura deterministas. A continuación se presentan versiones equivalentes sin variables de inventario, y se demuestra una de estas equivalencias. La demostración de las otras inecuaciones sigue un razonamiento análogo.

Proposición 10 *Las inecuaciones válidas para SELSRs (4.18),(4.19),(4.21),(4.22) se pueden escribir alternativamente sin variables de inventario como (4.23),(4.24),(4.25),(4.26) respectivamente:*

$$\sum_{n \in \mathcal{P}(a(i))} (x_n^m + x_n^r) + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^m + y_n^r) \geq D_{1j} \quad j \in N, i \in \mathcal{P}(j) \quad (4.23)$$

$$\sum_{n \in \mathcal{P}(a(i))} x_n^m + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^m) \geq D_{1j} \quad j \in N, i \in \mathcal{P}(j) \quad (4.24)$$

$$\sum_{n \in \mathcal{P}(a(i))} x_n^r + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^r) \geq D_{1j} \quad j \in N, i \in \mathcal{P}(j) \quad (4.25)$$

$$\sum_{n \in \mathcal{P}(j)} x_n^r - \sum_{n \in \mathcal{P}(i,j)} R_{in} y_n^r \leq R_{1a(i)} \quad j \in N, i \in \mathcal{P}(j) \quad (4.26)$$

Prueba

A continuación se demuestra la equivalencia entre las formulaciones (4.18) y (4.23). La demostración de las otras inecuaciones sigue un razonamiento análogo. Consideramos las ecuaciones de tipo (3.2) de la formulación natural para los nodos pertenecientes a un camino desde el nodo raíz, a un nodo k .

$$I_1^s + d_1 = x_1^m + x_1^r,$$

$$I_2^s + d_2 = I_{a(2)}^s + x_2^m + x_2^r = I_1^s + x_2^m + x_2^r$$

⋮

$$I_k^s + d_k = I_{a(k)}^s + x_k^m + x_k^r,$$

Si sumamos todas las ecuaciones, se observa que todas las variables de inventario se cancelan, a excepción de la última, $I_{a(k)}^s$, obteniéndose la siguiente ecuación:

$$I_k^s = \sum_{n \in \mathcal{P}(1,k)} (x_n^m + x_n^r) - D_{1k} \quad (4.27)$$

Consideremos la ecuación (4.18):

$$I_{a(i)}^s + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^m + y_n^r) \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j)$$

Realizando el cambio de variable (4.27) con $k = a(i)$ en la inecuación anterior se obtiene:

$$\sum_{n \in \mathcal{P}(a(i))} (x_n^m + x_n^r) - D_{1a(i)} + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^m + y_n^r) \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j)$$

Sumamos a ambos miembros de la inecuación el término $D_{1a(i)}$:

$$\sum_{n \in \mathcal{P}(a(i))} (x_n^m + x_n^r) + \sum_{n \in \mathcal{P}(i,j)} D_{nj}(y_n^m + y_n^r) \geq D_{ij} + D_{1a(i)}, \quad j \in N, i \in \mathcal{P}(j)$$

Dado que $D_{ij} + D_{1a(i)} = D_{1j}$, la expresión anterior coincide con la inecuación (4.23) y queda demostrada la equivalencia.

QEP

4.2.2. Inecuaciones válidas para SELSRj

Para el SELSRj estudiamos la inclusión de dos familias de inecuaciones basadas en la Proposición 3. Las familias de inecuaciones pueden ser adaptadas a la variante estocástica acorde a la siguiente proposición:

Proposición 11 *Las siguientes inecuaciones son válidas para el SELSRj:*

$$I_{a(i)}^S + \sum_{n \in \mathcal{P}(i,j)} D_{nj} y_n \geq D_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.28)$$

$$I_j^R + \sum_{n \in \mathcal{P}(i,j)} R_{in} y_n \geq R_{ij}, \quad j \in N, i \in \mathcal{P}(j) \quad (4.29)$$

Prueba

El razonamiento para demostrar esta proposición, es análogo al de la Proposición 8, observando que cada inecuación de la familia, coincide con una inecuación de tipo (I, S, WW) para cada escenario del árbol. Además, dicho escenario coincide con un problema de remanufactura determinista con una única línea de producción.

QEP

4.2.2.1. Inecuaciones sin variables de inventario SELSRj

Análogamente a lo realizado para la variante SELSRs, es posible reformular las dos familias de inecuaciones anteriores eliminando las variables de inventario. A continuación se presentan versiones equivalentes sin variables de inventario.

$$\sum_{n \in \mathcal{P}(a(i))} (x_n^m + x_n^r) + \sum_{n \in \mathcal{P}(i,j)} D_{nj} y_n \geq D_{1j} \quad j \in N, i \in \mathcal{P}(j) \quad (4.30)$$

$$\sum_{n \in \mathcal{P}(j)} x_n^r - \sum_{n \in \mathcal{P}(i,j)} R_{in} y_n \leq R_{1a(i)} \quad j \in N, i \in \mathcal{P}(j) \quad (4.31)$$

La demostración de estas equivalencias es muy similar a la detallada en la Sección 4.2.1.1, por lo que no se presenta en este informe.

4.3. Restricciones de árbol

Un conjunto de inecuaciones válidas puede combinarse para formar una nueva inecuación válida, cuya incorporación a la formulación pueda posiblemente permitirnos acercarnos al caso convexo. Esta idea puede utilizarse en problemas estocásticos, donde se busca combinar inecuaciones que son válidas para distintos escenarios, y obtener nuevas que se denominan inecuaciones de árbol (Guan et al., 2006c).

Esta Sección comienza describiendo la operación de emparejamiento, a través de la cual se construye la estrategia de combinación que será utilizada. Luego, se expone un esquema para derivar una familia de inecuaciones de árbol para problemas genéricos, partiendo de un conjunto de inecuaciones válidas de camino, tal como es descrito en el trabajo de Guan et al. (2006b). Se concluye la Sección con la deducción y estudio de inecuaciones de árbol para uno de los problemas estudiados en este trabajo, el SELSRs, dado que es la variante más compleja de las dos consideradas. Retel Helmrich et al. (2014) destacan que en la práctica la resolución del problema determinista con líneas de producción separadas es más difícil de resolver que el problema con una única línea de producción (ELSRj). Por lo tanto, se seleccionó este problema, observándose que la deducción para el SELSRj es similar.

4.3.1. Emparejamiento

Dados dos vectores a_1 y a_2 de igual dimensión, para los que se definen las operaciones $\min\{a_1, a_2\}$ y $\max\{a_1, a_2\}$ de modo que la minimización y maximización se realizan componente a componente. Dado un vector a y un escalar b , se define $a + b = a + b\mathbb{1}$ y $\min\{a, b\} = \min\{a, b\mathbb{1}\}$, donde $\mathbb{1}$ es un vector de unos de la misma dimensión que a .

Teorema 1 (Guan et al., 2007) *Dados los vectores $\hat{x} \in \mathbb{R}_+^p$, $\hat{y} \in \mathbb{Z}_+^n$. Dadas las inecuaciones $g_1\hat{x} + a_1\hat{y} \geq b_1$ y $g_2\hat{x} + a_2\hat{y} \geq b_2$ con $b_1 \leq b_2$. Si las inecuaciones son válidas para el conjunto $X \subset \mathbb{R}_+^p \times \mathbb{Z}_+^n$, entonces la inecuación emparejada $\varphi\hat{x} + \phi\hat{y} \geq b_2$ también es válida para X , donde*

$$\varphi = \max\{g_1, g_2\} \text{ y } \phi = \min\{a_1 + (b_2 - b_1), \max\{a_1, a_2\}\}$$

La operación de emparejamiento se puede realizar iterativamente, tomando como base inecuaciones ya emparejadas para luego repetir la operación y obtener nuevas inecuaciones válidas. Utilizando esta idea, es posible desarrollar una estrategia de combinación de inecuaciones de camino para obtener inecuaciones de árbol. Dicho procedimiento se describe en la siguiente subsección.

4.3.2. Restricciones de árbol para problemas estocásticos generales

En el trabajo de Guan et al. (2006b), se presenta un esquema general para derivar una familia de inecuaciones válidas a nivel del árbol, partiendo de un conjunto de inecuaciones válidas de camino.

Consideremos un problema de programación estocástica genérico como el presentado en la Sección 2.4.1.2. Dado que el problema es genérico, cualquier resultado que se desprenda de la aplicación del Teorema 1 sobre esta formulación, también será válida para los problemas SELSRs y SELSRj. Tomando como ejemplo el SELSRs, se puede analizar a cuales variables de este problema se corresponden las variables de la formulación genérica. El vector \hat{x}_j de la formulación en la Sección 2.4.1.2 se corresponderá con las variables de producción e inventario en cada nodo j para SELSRs. Análogamente, el vector \hat{y}_j se corresponderá con las variables de activación de producción en el nodo j para SELSRs, y su dominio se adaptará para tener variables binarias, $\hat{y}_j \in \mathbb{B}^n$.

Se agrega la siguiente notación para la formulación (2.39)-(2.41) de la Sección 2.4.1.2. Consideramos un conjunto de nodos R con cardinalidad K , tal que $R \subseteq \mathcal{V}$. La cardinalidad de R será la cantidad de inecuaciones que se desea emparejar. Sea $\mathcal{V}_R = \bigcup_{i \in R} \mathcal{P}(i)$ y $\mathcal{V}(j)$ el subárbol formado por los descendientes de j , se define $R(j) = R \cap \mathcal{V}(j)$ para cada $j \in \mathcal{V}_R$. Para cada nodo i del conjunto R , consideramos la inecuación válida:

$$\sum_{j \in \mathcal{P}(i)} (g_{ij}\hat{x}_j + a_{ij}\hat{y}_j) \geq b_i \quad (4.32)$$

Se asume que los coeficientes de las inecuaciones son siempre positivos y que el conjunto R se puede ordenar acorde al valor de los términos independientes, $R = \{i_1, \dots, i_K\}$, donde $0 =: b_{i_0} \leq b_{i_1} \leq b_{i_2} \leq \dots \leq b_{i_K}$. Correspondiendo a este conjunto de inecuaciones de camino, se definen parámetros para generar las inecuaciones de árbol:

$$b(R) = b_{i_K}, \quad (4.33)$$

$$\Delta_j(R) = \sum_{i_k \in R(j)} (b_{i_k} - b_{i_{k-1}}) \quad \forall j \in \mathcal{V}_R, \quad (4.34)$$

$$\varphi_j(R) = \max_{i \in R} \{g_{ij}\} \quad \forall j \in \mathcal{V}_R, \quad (4.35)$$

$$\phi_j(R) = \min\{\max_{i \in R} \{a_{ij}\}, \Delta_j(R)\} \quad \forall j \in \mathcal{V}_R. \quad (4.36)$$

Teorema 2 (Guan et al., 2006b) *Dado $R \subseteq \mathcal{V}$, e inecuaciones válidas como en (4.32), entonces la inecuación de árbol siguiente es válida para la región factible del problema de la Sección 2.4.1.2:*

$$\sum_{j \in \mathcal{V}_R} (\varphi_j(R)\hat{x}_j + \phi_j(R)\hat{y}_j) \geq b(R)$$

El Teorema 2 se puede demostrar mediante inducción, en base a la utilización iterativa de la operación de emparejamiento sobre las restricciones de camino y puede encontrarse en Guan et al. (2006b). Haciendo uso de este teorema, en la siguiente sección se deducen inecuaciones de árbol para algunas de las familias de inecuaciones de la Sección 4.2. Luego, se mencionan algunas estrategias para su inclusión en la formulación y para la implementación computacional.

4.3.3. Restricciones de árbol para SELSRs

Utilizando el Teorema 2, es posible combinar inecuaciones para el problema SELSRs. Como se mencionó en la subsección anterior, el vector \hat{x}_j de la formulación en 2.4.1.2 se corresponderá con las variables de producción e inventario en cada nodo j para SELSRs. El vector \hat{y}_j se corresponderá con las variables de activación de producción en el nodo j para SELSRs, y su dominio se cambiará a $\hat{y}_j \in \mathbb{B}^n$.

Para poder desarrollar una reformulación del problema con el agregado de inecuaciones para el problema SELSRs, previamente se deben tomar algunas decisiones. En primer lugar, se debe elegir la familia o familias de restricciones que se usarán. En segundo lugar, se debe decidir de a cuantas restricciones se realizará la combinación, es decir, la cardinalidad del conjunto R . Finalmente, también se debe decidir las combinaciones específicas que se agregarán a la formulación.

Para realizar la deducción de restricciones de árbol, se seleccionó (4.18). A diferencia de lo que sucede para las inecuaciones (4.19) a (4.22), en dicha inecuación participan los dos tipos de variables binarias y^m e y^r . Por simplicidad, se utiliza conjuntos R de cardinalidad 2, es decir, las inecuaciones se combinan de a pares. En este sentido, también nos apoyamos en los resultados del trabajo de Guan et al. (2006c), quienes obtienen mejoras importantes con la combinación de inecuaciones de a pares para un problema estocástico de producción sin remanufactura.

Deducción para la familia (4.18) del SELSRs

Analizamos en esta sección el armado de inecuaciones de árbol para restricciones de la familia (4.18) de la formulación del SELSRs. En esta familia participan los dos tipos de variables binarias y^m e y^r , lo cual no sucede en las otras familias estudiadas. Al seleccionar esta familia de inecuaciones, no quedan variables binarias del problema excluidas. Se debe tener en cuenta que para identificar una restricción dentro de la familia, se deben elegir dos nodos. Por lo tanto, a diferencia de lo que sucede en el trabajo de Guan et al. (2006b), los conjuntos R tendrán como elementos pares de nodos. A su vez, se busca combinar inecuaciones de a pares. Por lo tanto, para seleccionar dos restricciones a emparejar, se debe seleccionar un total de cuatro nodos.

Dados el par de nodos (i_1, j_1) correspondientes a una restricción de la familia (4.18) y el par (i_2, j_2) correspondiente a otra restricción de la misma familia, que cumplen las siguientes propiedades:

- $D_{i_1 j_1} \neq D_{i_2 j_2}$
- $j_2 \notin \mathcal{P}(j_1)$ y $j_1 \notin \mathcal{P}(j_2)$

El conjunto R estará dado por, $R = \{(i_1, j_1), (i_2, j_2)\}$ y se define $R' = \{j_1, j_2\}$. Dado que en las restricciones no participan nodos por fuera de los caminos $\mathcal{P}(i_1, j_1), \mathcal{P}(i_2, j_2)$, se define $\mathcal{V}_R = \bigcup_{(i,j) \in R} \mathcal{P}(i, j)$ y $R(j) = \{(i, j) \mid j \in R' \cap \mathcal{V}(j)\}$ Los valores de los parámetros para formar las restricciones de árbol son los siguientes:

$$b(R) = \max\{D_{i_1 j_1}, D_{i_2 j_2}\} \quad (4.37)$$

$$\Delta_n(R) = \begin{cases} \max\{D_{i_2 j_2}, D_{i_1 j_1}\}, & \text{si } n \in \mathcal{P}(j_1), n \in \mathcal{P}(j_2), n \in \mathcal{V}_R \\ D_{i_1 j_1} - D_{i_2 j_2}, & \text{si } n \in \mathcal{P}(j_1), n \notin \mathcal{P}(j_2), D_{i_1 j_1} \geq D_{i_2 j_2}, n \in \mathcal{V}_R \\ D_{i_1 j_1}, & \text{si } n \in \mathcal{P}(j_1), n \notin \mathcal{P}(j_2), D_{i_1 j_1} < D_{i_2 j_2}, n \in \mathcal{V}_R \\ D_{i_2 j_2} - D_{i_1 j_1}, & \text{si } n \notin \mathcal{P}(j_1), n \in \mathcal{P}(j_2), D_{i_1 j_1} \leq D_{i_2 j_2}, n \in \mathcal{V}_R \\ D_{i_2 j_2}, & \text{si } n \notin \mathcal{P}(j_1), n \in \mathcal{P}(j_2), D_{i_1 j_1} > D_{i_2 j_2}, n \in \mathcal{V}_R \end{cases} \quad (4.38)$$

$$\varphi_n(R) = \begin{cases} 0, & \text{si } n \neq i_1, n \neq i_2, n \in \mathcal{V}_R \\ 1, & \text{si } n = i_1, n \in \mathcal{V}_R \\ 1, & \text{si } n = i_2, n \in \mathcal{V}_R \end{cases} \quad (4.39)$$

$$\phi_n(R) = \min\{\Delta_n(R), \max\{D_{n, j_1}, D_{n, j_2}\}\} \quad \forall n \in \mathcal{V}_R, \quad (4.40)$$

El valor de (4.37) se obtiene de observar que las demandas acumuladas son los términos independiente de las restricciones (4.18). Para obtener el valor de (4.38) primero se debe identificar la posición del nodo n en el árbol con respecto a los caminos $\mathcal{P}(j_1)$ y $\mathcal{P}(j_2)$, obteniéndose tres posibilidades: n se encuentra en ambos caminos, n se encuentra sólo en $\mathcal{P}(j_1)$, y n se encuentra sólo en $\mathcal{P}(j_2)$. Luego, se debe determinar cuál término independiente tiene mayor valor y finalmente se opera acorde a lo explicado en el Teorema 2. El término (4.39) se obtiene observando que las únicas variables reales son las variables de inventario, y éstas aparecen una única vez en cada restricción. Finalmente, el término (4.40) se desprende de observar los coeficientes de las variables enteras y^m e y^r y realizar la minimización acorde al Teorema 2.

Aplicando el Teorema 2 y los parámetros explicados anteriormente, obtenemos las restricciones de árbol siguiente según los valores de $a(i_1)$ y $a(i_2)$:

$$I_{a(i_1)}^s + I_{a(i_2)}^s + \sum_{n \in \mathcal{P}(i_1, j_1) \cup \mathcal{P}(i_2, j_2)} \phi_n(R)(y_n^m + y_n^r) \geq b(R), \quad \text{si } a(i_1) \neq a(i_2) \quad (4.41)$$

$$I_{a(i_1)}^s + \sum_{n \in \mathcal{P}(i_1, j_1) \cup \mathcal{P}(i_2, j_2)} \phi_n(R)(y_n^m + y_n^r) \geq b(R), \quad \text{si } a(i_1) = a(i_2) \quad (4.42)$$

Consideraciones de implementación para la familia (4.18) del SELSRs

La deducción explicada en la Sección anterior nos permite combinar dos restricciones y generar una inecuación de árbol. Sin embargo, no existen garantías de que la selección de dos inecuaciones elegidas aleatoriamente logren fortalecer nuestra formulación del problema. Por lo tanto, resulta conveniente combinar una mayor cantidad de inecuaciones.

Al aplicar esta estrategia, se debe tener cuidado que el agregado de una cantidad muy grande de inecuaciones puede impactar negativamente los tiempos de resolución del solver de optimización. Idealmente, se puede realizar un estudio teórico de dominancia de inecuaciones, de modo de sólo agregar aquellas que se tiene certeza agregarán valor, ó utilizar alguna otra estrategia de gestión de cortes. No obstante, en el trabajo de Guan et al. (2006c), donde se estudia un problema de producción estocástico sin remanufactura, se reporta que pese a agregar una cantidad muy grande de inecuaciones, la experimentación utilizando mecanismos de gestión de cortes no arrojó mejoras importantes.

De modo de limitar los pares de inecuaciones que se combinan en nuestro problema, se utiliza una técnica similar a la estrategia de aproximación parcial propuesta por Vyve (2006).

Esta técnica fue utilizada en formulaciones deterministas de gran tamaño de modo de lograr que sean abordables computacionalmente.

Como resultado de un estudio experimental preliminar, se comprobó que a pesar de utilizar diversas combinaciones con la técnica descrita anteriormente, el tiempo de armado del modelo es prohibitivamente alto para instancias de ocho o más períodos. Para comprender el motivo es necesario analizar algunos aspectos de la implementación computacional de las inecuaciones. Siguiendo la técnica de aproximación parcial, se implementa una familia de restricciones donde se asigna uno o más rangos de valores a los nodos i_1, i_2, j_1, j_2 . Para implementar dicha familia debe existir una estructura de datos para los parámetros presentes en las restricciones (4.41) y (4.42). Para el cálculo de ϕ , es necesario contar con los valores de demandas acumuladas entre dos nodos. Por lo tanto, esta estructura debe contener todos los valores posibles del parámetro para todo el rango de valores de los nodos i_1, i_2, j_1, j_2 , y para nodos n , con $n \in \mathcal{V}_R$. Dado que para calcular el valor del parámetro se debe hacer el análisis visto anteriormente, la estructura de datos necesariamente tendrá cinco valores de entrada. Sean I_1, I_2, J_1, J_2 conjuntos a partir de los que se seleccionan los valores i_1, i_2, j_1, j_2 . La estructura de datos ϕ entonces deberá contener al menos $|I_1| \times |I_2| \times |J_1| \times |J_2|$ valores de entrada diferentes, a los que se debe agregar los valores que puede tomar n , que son los nodos de los caminos de todas las restricciones que se vayan a combinar. De esta manera, si se utiliza un rango amplio para los valores de entrada, la estructura de datos que se utiliza en la implementación para poder representar todos los posibles valores de ϕ puede tener un tamaño muy grande. Así, el tiempo de armado del modelo por parte de la herramienta puede ser elevado, ya que se debe crear una estructura de datos de un tamaño considerable.

En busca de solucionar esta problemática, realizamos un deducción similar a la presentada en esta sección sobre la familia de restricciones (4.23). Como fue visto en la Sección 4.2, esta familia es equivalente a (4.18) en cuanto a las relajaciones a programación lineal de las formulaciones con una u otra familia. Sin embargo, la familia (4.23) tiene como término independiente demandas acumuladas desde el nodo raíz hasta un nodo, a diferencia de (4.18) cuyo término independiente es la demanda acumulada entre dos nodos. En la siguiente sección se verá que por este motivo, el cálculo de ϕ depende de una menor cantidad de nodos. De esta manera, el parámetro ϕ se podrá implementar tomando una menor cantidad de valores de entrada, disminuyendo así su tamaño y el tiempo de construcción del modelo.

Deducción para la familia (4.23) del SELSRs

Analizamos en esta sección el armado de inecuaciones de árbol para restricciones de la familia (4.23) de la formulación del SELSRs. En este caso, a diferencia de lo que sucede con la familia (4.18), los términos independientes de las restricciones son demandas acumuladas que parten del nodo raíz. Este aspecto permite que la implementación computacional de los parámetros que participan en las restricciones de árbol tenga una estructura de datos menor, ya que su valor depende de una menor cantidad de nodos. Por este motivo, los conjuntos R los definiremos de modo que su cardinalidad sea dos y sus elementos serán nodos individuales, a diferencia de los pares de nodos de la familia (4.18).

Al igual que en la sección anterior, se busca combinar inecuaciones de a pares. Para calcular el valor de los parámetros que participan en las restricciones de árbol, basta ahora seleccionar un total de dos nodos. Dados los nodos j_1, j_2 de SELSRs, que cumplen las siguientes dos propiedades:

- $D_{1j_1} \neq D_{1j_2}$
- $j_2 \notin \mathcal{P}(j_1)$ y $j_1 \notin \mathcal{P}(j_2)$

El conjunto R estará dado por, $R = \{j_1, j_2\}$. Se define $\mathcal{V}_R = \bigcup_{i \in R} \mathcal{P}(i)$, y se define $R(j) = R \cap \mathcal{V}(j)$ para cada $j \in \mathcal{V}_R$. Los valores de los parámetros para formar las restricciones de árbol son los siguientes:

$$b(R) = \max\{D_{1j_1}, D_{1j_2}\} \quad (4.43)$$

$$\Delta_n(R) = \begin{cases} \max\{D_{1j_2}, D_{1j_1}\}, & \text{si } n \in \mathcal{P}(j_1), n \in \mathcal{P}(j_2), n \in \mathcal{V}_R \\ D_{1j_1} - D_{1j_2}, & \text{si } n \in \mathcal{P}(j_1), n \notin \mathcal{P}(j_2), D_{1j_1} \geq D_{1j_2}, n \in \mathcal{V}_R \\ D_{1j_1}, & \text{si } n \in \mathcal{P}(j_1), n \notin \mathcal{P}(j_2), D_{1j_1} < D_{1j_2}, n \in \mathcal{V}_R \\ D_{1j_2} - D_{1j_1}, & \text{si } n \notin \mathcal{P}(j_1), n \in \mathcal{P}(j_2), D_{1j_1} \leq D_{1j_2}, n \in \mathcal{V}_R \\ D_{1j_2}, & \text{si } n \notin \mathcal{P}(j_1), n \in \mathcal{P}(j_2), D_{1j_1} > D_{1j_2}, n \in \mathcal{V}_R \end{cases} \quad (4.44)$$

$$\varphi_{n,i_1,i_2}(R) = \begin{cases} 1, & \text{si } n \in \mathcal{P}(i_1) \cup \mathcal{P}(i_2), n \in \mathcal{V}_R \\ 0, & \text{otro caso} \end{cases} \quad (4.45)$$

$$\phi_n(R) = \min\{\Delta_n(R), \max\{D_{n,j_1}, D_{n,j_2}\}\} \quad \forall n \in \mathcal{V}_R, \quad (4.46)$$

El valor (4.43) se obtiene de observar que las demandas acumuladas desde la raíz a un nodo j es el término independiente de las restricciones (4.23). Para obtener el valor de (4.44) primero se debe identificar la posición del nodo n en el árbol con respecto a los caminos

$\mathcal{P}(j_1)$ y $\mathcal{P}(j_2)$, obteniéndose tres posibilidades: n se encuentra en ambos caminos, n se encuentra sólo en $\mathcal{P}(j_1)$, n se encuentra sólo en $\mathcal{P}(j_2)$. Luego, se debe determinar cuál término independiente tiene mayor valor y finalmente se opera acorde a lo explicado en el Teorema 2. Por otro lado, (4.45) se obtiene observando que las únicas variables reales son las variables de producción, y éstas aparecen en las restricciones cuando están en alguno de los caminos $\mathcal{P}(i_1)$, $\mathcal{P}(i_2)$. El último parámetro, (4.46), se desprende de observar los coeficientes de las variables enteras y^m e y^r y realizar la minimización acorde al Teorema 2.

Aplicando el Teorema 2 y los parámetros explicados anteriormente, obtenemos la restricción de árbol siguiente:

$$\sum_{n \in \mathcal{P}(a(i_1)) \cup \mathcal{P}(a(i_2))} (x_n^m + x_n^r) + \sum_{n \in \mathcal{P}(i_1, j_1) \cup \mathcal{P}(i_2, j_2)} \phi_n(R)(y_n^m + y_n^r) \geq b(R) \quad (4.47)$$

Consideraciones de implementación para la familia (4.23) del SELSRs

La deducción explicada en la Sección anterior nos permite combinar dos restricciones y generar una inecuación de árbol. Para fortalecer la formulación, se buscará combinar una mayor cantidad de inecuaciones.

Análogamente a lo explicado en la Sección 4.3.3, de modo de limitar los pares de inecuaciones que se combinan en nuestro problema, se utiliza una técnica similar a la técnica de aproximación parcial propuesta por Vyve (2006).

Se implementa una nueva familia de restricciones donde se asigna rangos de valores a los nodos i_1, i_2, j_1, j_2 . Para implementar dicha familia debe existir una estructura de datos para el parámetro ϕ . Esta estructura debe contener todos los valores posibles del parámetro para todo el rango de valores de los nodos j_1, j_2 , y para nodos n , con $n \in \mathcal{V}_R$. Así, el parámetro ϕ se pudo implementar tomando una menor cantidad de valores de entrada, disminuyendo así su tamaño. Este concepto se puede reafirmar de la siguiente manera. Sean J_1, J_2 conjuntos a partir de los que se seleccionan los valores j_1, j_2 . La estructura de datos ϕ entonces deberá contener $|J_1| \times |J_2|$ valores de entrada, a los que se debe agregar los valores que puede tomar n , que se obtienen de manera análoga a lo consideraciones de implementación para la familia (4.18). El tamaño de la estructura ϕ podrá tener una menor cantidad de valores de entrada en (4.23) en comparación con (4.18). Como resultado de un estudio experimental preliminar utilizando la técnica mencionada, se determinó que el tiempo de armado del modelo fue

considerablemente menor que utilizando las inecuaciones (4.41) y (4.42).

Además, se corroboró que es posible mejorar las relajaciones a programación lineal de las formulaciones a través de la incorporación de las restricciones de árbol (4.47) a la formulación con restricciones de camino. Este resultado se pudo obtener utilizando la técnica mencionada, con una misma selección de rangos sobre distintas instancias. De esta manera, se pudo verificar que la combinación de restricciones de camino tiene el potencial de mejorar las formulaciones con inecuaciones válidas para el SELSRs. Por otro lado, no se consiguió mejorar la calidad de las soluciones obtenidas al ejecutar las formulaciones con un límite de tiempo dado. Esto se puede deber a que la técnica propuesta requiere de la incorporación de una cantidad de inecuaciones para mejorar las relajaciones que impacta negativamente el desempeño del solver. En el Capítulo 8 se presentan posibles trabajos futuros, que permitan desarrollar mecanismos más sofisticados que la técnica mencionada en esta sección para incorporar restricciones de árbol, y así poder ser utilizadas en la práctica.

Capítulo 5

Resolución Heurística

Los métodos de resolución exacta, garantizan que en caso de que exista una solución óptima, se llegará a la misma si se brinda al método los recursos computacionales suficientes y el tiempo necesario de ejecución. A partir de determinados tamaños de instancia, la capacidad de cómputo necesario puede ser muy superior a la disponible. Los métodos de resolución heurística, en contrapartida, requieren una capacidad de cómputo menor a las estrategias de resolución exacta, pero no dan garantías de la calidad de las soluciones encontradas. En el Capítulo anterior se investigaron diversos mecanismos de resolución exactos, asociadas al fortalecimiento de las formulaciones de los problemas SELSRs y SELSRj. En el estudio computacional, estas formulaciones se utilizarán agregando criterios de parada, y seleccionando restricciones a agregar con el objetivo de obtener buenas soluciones en menor cantidad de tiempo. Habiendo presentado los principales aspectos asociadas a una herramienta de resolución exacta, en este capítulo se investiga una metodología de resolución heurística computacional para el SELSRs. De esta manera, en el Capítulo se presentan los detalles de la selección y diseño de una aproximación heurística, así como los distintos aspectos de la implementación computacional.

5.1. Descripción del método relax-and-fix

Como fue visto en el Capítulo 2, los problemas de planificación de producción han sido abordados mediante una gran cantidad de heurísticas tradicionales, como son la búsqueda tabú, los algoritmos genéticos, entre otros. Para el caso particular de problemas con remanufactura, mucho de estos mecanismos de resolución han podido ser adaptados de versiones clásicas. Teunter et al. (2006) analizan el desempeño de las heurísticas Silver Meal, Least Unit Cost y Part Period Balancing sobre las variantes ELSRj y ELSRs sin costos de producción unitarios. Van den Heuvel (2006) aplican algoritmos genéticos y programación dinámica para resolver

versiones simplificadas del ELSRs en las que los períodos donde se activa la producción son conocidos. Piñeyro y Viera (2010) evalúan una heurística de búsqueda Tabú en el que la demanda de productos remanufacturados se puede satisfacer con nuevos productos, pero no a la inversa. A su vez, se permite la opción de deshechar retornos.

Para variantes con remanufactura estocásticos, la literatura que trata heurísticas computacionales para resolver la problemática es escasa. La adaptación de estas heurísticas, tradicionales para el problema clásico, como por ejemplo la búsqueda Tabú, presenta desafíos importantes. Para la búsqueda Tabú, el diseño de los movimientos para trasladarse de una solución a otra no tiene un diseño intuitivo como para las variantes deterministas. Por otro lado, la heurística relax-and-fix es basada en programación matemática, lo que permite aplicarla a nuestra formulación de una manera más directa.

Roshani et al. (2017) utilizan la heurística relax-and-fix para resolver una versión determinista y capacitada que permite satisfacer la demanda remanufacturando retornos. Beraldi et al. (2006) utiliza la heurística relax-and-fix para resolver un problema en el que se debe asignar la producción de varios ítems en una cierta cantidad de máquinas. La estrategia de resolución se basa en el trabajo de Alonso et al. (2000), quienes desarrollan una heurística relax-and-fix para un problema de manejo de tráfico aéreo.

Las heurísticas de tipo relax-and-fix buscan resolver iterativamente, mediante un solver de optimización, una serie de problemas de menor tamaño. En este método, se relajan las restricciones de integralidad de algunas de las variables, y se fijan con el valor de ejecuciones de iteraciones previas otras de las variables enteras/binarias. De este modo, se reduce la cantidad de variables enteras/binarias, con lo que se espera obtener problemas más sencillos de resolver (Beraldi et al., 2006; Roshani et al., 2017). Para entender por que esto puede suceder, es preciso tener en cuenta el funcionamiento del solver. El método branch-and-bound genera un árbol de subproblemas activos, el que en el peor de los casos debe analizar el valor de cada una de las variables enteras/binarias. Como se tiene una menor cantidad de variables discretas, el tamaño del árbol será menor, y la cantidad de opciones a analizar también lo será.

Dado el conjunto de variables binarias del problema original, en cada iteración seleccionaremos:

- Un subconjunto de variables a relajar, de modo que su dominio será un rango de los números reales. Dado que el problema SELSRs define variables binarias, la relajación

de la integralidad de dichas variables se realiza permitiendo que las mismas tomen cualquier valor real en el rango entre 0 y 1.

- Un subconjunto de variables a fijar con resultados de iteraciones anteriores.
- Un subconjunto de variables para las que se mantiene el dominio original.

Este enfoque de resolución se puede adaptar para resolver un problema estocástico, como es el caso del SELSRs. El objetivo es resolver una secuencia de subproblemas asociados a los nodos del árbol de escenarios. Un procedimiento general fue descrito formalmente en Beraldi et al. (2006) para un problema de secuenciamiento con variables enteras. A continuación se presenta el procedimiento aplicable a problemas de dimensionado de lotes estocásticos con variables binarias. Dado el conjunto de nodos del árbol V , consideramos la partición del mismo en t subconjuntos V_t , siendo t un período del horizonte de planificación. Así, los nodos de una partición pertenecen todos a un mismo período. Se resuelven k subproblemas P^r , con $r = 1, \dots, k$. El orden de resolución parte de $r = 1$ en orden ascendente hasta $r = k$, y se debe corroborar que se mantiene la factibilidad en cada paso. Las restricciones de integralidad se modifican acorde al siguiente criterio:

- Se fijan las variables binarias del nodo n para $n \in V_p$, $p = 1, \dots, r - 1$ (si $r > 1$)
- Se mantiene la integralidad de las variables del nodo n para $n \in V_r$
- Se relaja la integralidad de las variables del nodo n para $n \in V_p$, $p = r + 1, \dots, k$ (si $r < k$)

En cada iteración se resuelven subproblemas utilizando los subconjuntos de variables definidos. Si la cantidad de variables relajadas en cada iteración es suficiente, se espera poder resolver el problema a optimalidad. Existen diversas maneras de seleccionar los subconjuntos en cada iteración. Una estrategia adecuada dependerá tanto de entender las características del problema estudiado, como de la experimentación, a través de la cual se pueden ajustar en mayor medida los parámetros de la heurística. En este trabajo, se investigaron tres estrategias. En la Sección 5.2, se presenta una implementación que divide los subconjuntos de variables acorde al período al que pertenecen. Para este caso, la cantidad de períodos al que pertenecen el subconjunto de variables que mantienen la restricción de dominio binario es constante entre las iteraciones. En la Sección 5.3 se presenta la descomposición en subproblemas dentro de un período de tiempo. Se presentan las dificultades que este esquema presenta para el caso particular del SELSRs, y su resolución. Finalmente, en la búsqueda de obtener un mejor

desempeño en la heurística, en la Sección 5.4 se presenta una implementación que modifica el enfoque de 5.2 y se adapta al problema estudiado.

5.2. Esquema relax-and-fix tradicional

Las heurísticas relax-and-fix al ser aplicadas sobre problemas de planificación de producción normalmente dividen el horizonte de planificación en tres ventanas. En esta sección se presenta un resumen del esquema de Roshani et al. (2017), que utiliza tamaños de ventana central constantes, y cuya generalidad lo vuelve aplicable a una gran cantidad de problemas de planificación de la producción. Luego, se presentan algunas consideraciones y dificultades observadas luego de experimentar con distintas implementaciones de esta estrategia sobre nuestro caso de estudio.

Se define una ventana inicial que comienza en el primer período y abarca los períodos comprendidos hasta un período k , una ventana central desde $k + 1$ hasta un período t , y una ventana final desde $t + 1$ hasta el último período. En la ventana inicial, el valor de las variables binarias se fijan al valor que tomaron en una iteración previa. En la ventana central, se mantiene el dominio de las variables del problema original. En la ventana final, las variables se relajan. En el caso de tratarse de variables binarias, las variables relajadas pueden tomar cualquier valor real entre 0 y 1.

Siguiendo el procedimiento descrito, se selecciona un valor fijo para el tamaño de la ventana central, y se selecciona un parámetro γ que representa el avance de las ventanas inicial y central en cada iteración. El valor de la ventana central y el incremento γ puede coincidir o no. En caso de coincidir, cada variable mantendrá el dominio original, es decir será binaria, una única vez. En caso que el incremento sea menor al tamaño de la ventana central, existirán variables que pertenezcan a la ventana central en más de una ocasión. A modo de ejemplo, si el tamaño de la ventana central es de tres períodos, y el incremento es de dos períodos, el tercer período pertenecerá a la ventana central tanto en la primer como en la segunda iteración.

Al empezar, la ventana central comienza en el primer período, y termina en un período acorde al valor fijo que se decidió para el tamaño de la ventana. Así, en la primer iteración, solamente se tiene la ventana central y final, ya que no hay iteraciones previas para fijar variables. El problema se resuelve a optimalidad. En la segunda iteración, teniendo la solución óptima del subproblema anterior, se toman las primer γ variables para formar la ventana inicial, es decir se fijan sus valores a los de la solución, y se ajustan las ventanas central y final acorde al

incremento γ . El problema se resuelve a optimalidad nuevamente. El procedimiento continúa de esta manera hasta tanto se llegue a una iteración en la que solo existen las ventanas inicial y central. La solución de esta iteración será finalmente la solución que devolverá la heurística.

Se implementó el método para tamaños de problemas entre 5 y 10 períodos sobre árbol de aridad 2 para el SELSRs. Como primer observación, se corroboró que la dificultad de resolución de los subproblemas depende en alta medida de la cantidad de variables binarias. En segundo lugar, no se observó beneficio al utilizar valores diferentes para el incremento y el tamaño de la ventana central. Para pruebas con una cantidad de períodos mayor a 6, y valores de ventana central mayores a 2, se observa que el método suele estancarse en iteraciones intermedias. Esto se debe a la gran cantidad de variables binarias que se deben considerar en estas iteraciones. Para resolver esta problemática, se estudió la modificación de agregar un límite de tiempo a cada iteración, y fijar los valores de las variables a partir de soluciones factibles pero posiblemente no óptimas de las iteraciones donde solía estancarse la heurística. Sin embargo, dada la estructura de árbol del problema, se decidió finalmente investigar otras estrategias: la descomposición en subproblemas dentro de un período, y la utilización de tamaños de ventana central variables.

5.3. Descomposición en subproblemas dentro de un período

Dado que los nodos son agrupados por período, algunos problemas pueden tener una cantidad de nodos muy elevada, haciendo imposible la resolución en tiempos prudenciales. Sin embargo, para algunos problemas es posible resolver el subproblema de una iteración descomponiendo el mismo en subproblemas independientes. Este es el caso Alonso et al. (2000) y Beraldi et al. (2006), quienes utilizaron heurísticas de tipo relax-and-fix para resolver problemas estocásticos.

Alonso et al. (2000) desarrollan una heurística relax-and-fix para un problema de manejo de tráfico aéreo en el que solamente se tienen variables binarias. El problema estudiado en Beraldi et al. (2006) consiste en determinar el secuenciamiento de la producción de varios items. De modo similar a otros problemas de secuenciamiento, se consideran dos tipos de variables: variables que denotan la cantidad de máquinas asignadas a un producto en un período, y variables que representan el número de máquinas que cambiaron la producción de un producto a otro dentro de un período. Dado que no hay variables continuas, el problema

es de Programación Entera.

Aprovechando la estructura de los problemas estudiados, los autores descomponen los problemas a resolver dentro de una iteración en problemas independientes, lo que disminuye drásticamente la cantidad de variables enteras a resolver. Existen diferencias importantes entre estos dos problemas y el SELSRs, en el que además de las variables binarias, se han definido variables continuas para la producción en cada nodo, y para los inventarios de productos listos y de retornos. A continuación se presenta un ejemplo que muestra el motivo por el cual esta descomposición no se puede realizar de esta manera en el SELSRs.

Consideremos un problema con un nodo raíz 1, y dos nodos en el segundo período 2 y 3, correspondientes a dos escenarios de demanda, en el que los costos de producir artículos nuevos son muy superiores a los de remanufacturar. Si la diferencia es suficientemente grande, convendrá satisfacer la demanda remanufacturando siempre que se pueda. Se tienen las siguientes demandas, costos de producción para remanufactura, costos de activar remanufactura y retornos, $d_1 = 1$, $d_2 = 2$, $d_3 = 1$, $p_1^r = 4$, $p_2^r = 1$, $p_3^r = 1$, $K_1^r = 1$, $K_2^r = 1$, $K_3^r = 1$, $r_1 = 2$, $r_2 = 1$, $r_3 = 1$. Cualquier solución factible debe activar la producción en el primer nodo, ya que hay demanda a satisfacer. En la primer iteración se decide activar la remanufactura. En la segunda iteración, se fija la activación en el primer nodo, y se intenta resolver los problemas de modo independiente. Se analiza el nodo 2, y se decide activar la producción. Además, se observa que dado que no hay suficientes retornos en 2, se puede remanufacturar un producto más en 1. Se analiza independientemente el nodo 3, y se observa que conviene activar la remanufactura, dado el alto costo de remanufactura en 1. En resumen, si descomponemos el problema, se activa la producción tanto en 2 como en 3. Sin embargo, existe una solución menos costosa, en la que se aprovecha el retorno sobrante de 1 para satisfacer tanto 2 como 3, y así, evitar la activación en 3.

Teniendo en cuenta la problemática planteada, analizamos dos posibles estrategias. Una posible estrategia es fijar en cada iteración no sólo las variables binarias, sino que también las variables continuas de nuestro problema, de modo de obtener problemas independientes en cada iteración. Otra posible estrategia surge de observar que la resolución independiente de los subproblemas si bien no necesariamente será equivalente a resolver el problema de la iteración como uno solo, puede brindarnos cotas similares. Por otro lado, se ajustó la definición de los conjuntos V_p de modo que no sólo consideren a un único período, sino que abarquen una mayor cantidad de períodos. Por ejemplo, los nodos de los períodos 1 a 5 se agruparon dentro del primer subproblema, ya que el tamaño de nodos es manejable por el

solver. Las primeras pruebas con ambas estrategias para instancias con pocos períodos no devolvieron buenos resultados, por lo que se optó por un tercer mecanismo de resolución.

5.4. Implementación relax-and-fix con ventana central variable

Considerando las limitaciones y ventajas de las dos estrategias vistas hasta este punto, se presenta a continuación una nueva implementación de la heurística relax-and-fix para el SELSRs. Como fue mencionado, la cantidad de variables binarias de un subproblema a resolver en una iteración de la heurística impacta de manera importante los tiempos de resolución. A su vez, la cantidad de nodos de un árbol en un período crece exponencialmente para árboles completos. Teniendo en cuenta estos dos aspectos, surge la motivación de explorar el funcionamiento de la heurística utilizando una ventana central cuyo tamaño varíe entre las distintas iteraciones.

Como fue visto en el Capítulo 3, en cada nodo se consideran dos variables binarias, una para activar la producción de artículos nuevos, y otra para activar la remanufacturación. Por lo tanto, la cantidad de variables binarias de los nodos de un período t para árboles 2-arios es 2^{t+1} . Realizando un estudio experimental, es posible determinar una cantidad máxima de variables binarias que pueden ser resueltas en determinada cantidad de tiempo. En este sentido, resulta importante destacar que existe otros aspectos que afectan los tiempos de resolución del subproblema, como la cantidad de variables continuas. Así, se tendrán tiempos de resolución algo más elevados en las resoluciones de programación lineal que se hacen en los nodos del branch-and-bound. Por ejemplo, se observa que el tiempo de ejecución en el procesamiento de un nodo del branch-and-bound tiene una diferencia no despreciable entre problemas de 8 y 12 períodos. Por otro lado, se observa que incluso en los casos en que la cantidad de variables binarias en una etapa coincide para 10 y 12 períodos, el tiempo de resolución de la etapa generalmente es mayor para las instancias de 12 períodos. Teniendo en cuenta esto, la experimentación debe considerar distintos tamaños de problema para determinar la cantidad máxima de variables binarias para cada iteración. Llamaremos δ al último período con el que se puede establecer la ventana central de modo de tener posibilidades de obtener soluciones óptimas en un tiempo de ejecución máximo fijado previamente. Dadas las capacidades de cómputo con las que se dispuso (ver Sección 6.2), se determinó que utilizando hasta 64 variables sobre la mayoría de las instancias probadas, se evita que la heurística quede estancada buscando soluciones en iteraciones intermedias. Este valor es dependiente del solver y la capacidad de cómputo con la que se cuenta, por lo que es de esperar que con

mayores recursos este valor pueda ser mayor.

Con este enfoque, se espera aumentar la probabilidad de obtener soluciones óptimas en las resoluciones previas a la ejecución final, en los períodos de tiempo establecidos para la heurística. Buscamos así implementaciones de la heurística que puedan variar la cantidad de iteraciones de modo de mantener la cantidad de variables binarias por debajo de dicho número. Para árboles 2-arios, y para la capacidad de cómputo con la que se cuenta, ejemplo de tamaños para la ventana central se pueden obtener hasta el sexto período.

En el Algoritmo se presenta un pseudocódigo de la implementación de la heurística. La misma se puede dividir en tres etapas. En una primer etapa, la heurística funciona con una ventana central de tamaño constante, de manera similar a la implementación presentada en la Sección 5.2. Esta etapa culmina una vez que la ventana central involucra nodos del período δ . La segunda etapa establece una ventana central únicamente con el período δ , ya que como se había visto, agregar una mayor cantidad de variables binarias puede dificultar la resolución. Finalmente, en una última etapa, se realiza una iteración con una ventana compuesta por todos los períodos restantes. La ejecución finalizará acorde a un criterio de parada de tiempo máximo. Por otro lado, se debe seleccionar la formulación a utilizar en cada subproblema para cada iteración. Se decidió utilizar la formulación con inecuaciones válidas derivadas en las Proposiciones 8 y 9 de la Sección 4.2.1 para la primer y segunda etapa dado el buen desempeño de la formulación para tamaños pequeños y medios de problema. Como se verá en el estudio computacional, la formulación con inecuaciones válidas se deteriora al considerar un tamaño de problema mayor a 10 períodos. Para tamaños de problemas con una gran cantidad de períodos, la tercer etapa consistirá en resolver un subproblema con una cantidad de restricciones muy elevada. En busca de disminuir la cantidad de restricciones, se optó por eliminar las restricciones válidas agregadas a la formulación natural durante la tercer etapa. Así, se espera que la heurística se pueda utilizar con mayores tamaños de problema.

Para implementar la heurística es necesario definir nuevos parámetros. Las ventanas inicial, central y final de cada iteración se pueden establecer utilizando parámetros que indiquen el comienzo y final de la ventana central. Se debe establecer un incremento para la ventana central. Este incremento aumentará el valor de los parámetros de la ventana central hasta que la misma llegue al último período en el que se puede obtener soluciones óptimas en un tiempo de ejecución preestablecido. Como fue mencionado, dicho valor debe ser calculado previamente, y se representa como δ . Dado que para tamaños grandes de problemas no se espera encontrar soluciones óptimas en la última iteración de la ejecución, se establece un

iteración	alfa	beta	iteración	alfa	beta	iteración	alfa	beta
1	1	5	1	1	3	1	1	2
2	6	6	2	4	5	2	3	4
3	7	12	3	6	6	3	5	6
			4	7	12	4	7	8
						5	9	9
						6	10	12

(a) $\gamma = 5, \delta = 6$ (b) $\gamma = 3, \delta = 6$ (c) $\gamma = 2, \delta = 9$

Tabla 5.1 Ejemplos de ejecución del pseudocódigo en los que se muestra, para tres combinaciones de valores de γ y δ , el primer y último período de la ventana central de la heurística en cada iteración.

criterio de parada de tiempo. Además, debemos tener una función auxiliar que calcule el período al que pertenece un nodo. Ejemplos de ejecución del pseudocódigo se presentan en la Tabla 5.1

α : primer período de la ventana central

β : último período de la ventana central

γ : incremento de α y β entre una iteración y la siguiente

δ : último período de ventana central en el que se puede obtener soluciones óptimas en un tiempo de ejecución definido anteriormente

te_{jec} : tiempo restante de ejecución de la heurística

$t(n)$: función auxiliar que calcula el período al que pertenece un nodo n

H : último período del horizonte de planificación

$etapa$: número de etapa (1,2,3) que se ejecuta

Algoritmo 1 relax-and-fix para SELSRs

Precondición: $\gamma < \delta$

- 1: **Pseudocódigo** RNF SELSRs
- 2: Inicializar γ , δ , $tejec$, y cargar $t(n)$
- 3: $\alpha \leftarrow 1$, $\beta \leftarrow \gamma$, $etapa \leftarrow 1$
- 4: Leer instancia
- 5: **while** ($etapa \leq 3$) **do**
- 6: Relajar variables binarias en n , para $n : t(n) > \beta$
- 7: **if** ($\alpha + \gamma = H + 1$) **then**
- 8: Eliminar inecuaciones válidas auxiliares
- 9: **end if**
- 10: Resolver subproblema con tiempo de ejecución $tejec$
- 11: $tejec \leftarrow (tejec - \text{tiempo de subproblema})$
- 12: Fijar variables en n , para $n : t(n) < \alpha + \gamma$
- 13: **if** ($etapa = 3$) **then**
- 14: $etapa \leftarrow etapa + 1$
- 15: **end if**
- 16: **if** ($etapa = 2$) **then**
- 17: $\alpha \leftarrow \delta + 1$
- 18: $\beta \leftarrow H$
- 19: $etapa \leftarrow etapa + 1$
- 20: **end if**
- 21: **if** ($etapa = 1$) **then**
- 22: $\alpha \leftarrow \alpha + \gamma$
- 23: **if** ($\alpha \geq \delta$) **then**
- 24: $\alpha \leftarrow \delta$
- 25: $\beta \leftarrow \delta$
- 26: $etapa \leftarrow etapa + 1$
- 27: **else if** ($\beta + \gamma \geq \delta$) **then**
- 28: $\beta \leftarrow \delta - 1$
- 29: **else**
- 30: $\beta \leftarrow \beta + \gamma$
- 31: **end if**
- 32: **end if**
- 33: **end while**
- 34: Extraer datos de solución de la última iteración
- 35: **end Pseudocódigo**

Capítulo 6

Estudio Computacional

En este capítulo se presenta el estudio computacional para analizar el desempeño de la formulación Natural (Capítulo 3), las reformulaciones (Capítulo 4) y la heurística (Capítulo 5) para el SELSRs. Este análisis se realiza mediante la ejecución de una gran cantidad de instancias de prueba, agrupadas en casos. Cada uno de estos casos corresponde a la selección de determinados valores para los parámetros más importantes del problema. Estos casos son a su vez presentados en distintas secciones acorde al tamaño del problema (pequeño, mediano, grande), ya que varios aspectos del problema, como su dificultad de resolución, dependen en gran medida de la cantidad de períodos en el horizonte de planificación. El desempeño de la heurística, en particular, es reservado para instancias del problema de tamaño mediano y grande, dado que para problemas de tamaño pequeño las estrategias de resolución exacta pueden devolver soluciones óptimas en tiempos de ejecución razonables.

En la Sección 6.1 se detallan aspectos técnicos de la generación automática de instancias. En la Sección 6.2, se presenta la selección de parámetros, la creación de casos de prueba, y la configuración de la experimentación numérica. En las Secciones 6.3, 6.4 y 6.5 se estudia el desempeño de las formulaciones presentadas para las instancias generadas para los casos con tamaño de problema pequeño, mediano y grande respectivamente.

6.1. Generación de casos de prueba

Se consideraron dos conjuntos de instancias de prueba para evaluar las versiones deterministas de los modelos presentados, el de Retel Helmrich et al. (2014) y el de Sifaleras et al. (2015). Hasta el momento se desconoce la existencia de un benchmark con casos de prueba para problema de dimensionamiento de lotes con remanufacturación e incertidumbre. Por lo tanto, resulta necesario generar casos de prueba para evaluar el desempeño de las

formulaciones presentadas.

Para problemas de planificación de producción que incorporan estocasticidad, la generación de casos de prueba manualmente no es viable, aún para casos con una pequeña cantidad de períodos. Además de haber una cantidad de nodos exponencial en el tamaño del horizonte de planificación, se debe tener en cuenta que las pruebas para los modelos planteados necesariamente deben contener toda la información referente a la estructura de los árboles de escenarios: caminos, subcaminos, hojas, etc. Por lo tanto, se decidió utilizar la herramienta A Mathematical Programming Language (AMPL) para la creación de un generador automático de instancias. AMPL es un lenguaje de modelado algebraico, y tiene la característica de tener una sintaxis muy similar a la utilizada en álgebra.

Las instancias se generan acorde a la especificación de una combinación de parámetros en un archivo de texto, en el que se debe explicitar:

- cantidad de períodos
- aridad del árbol
- problema para el cual se desea generar instancias (líneas de producción común o separadas)
- valores de media y varianza para la demanda
- valores de media y varianza para los retornos
- valores de costos
- probabilidad de ocurrencia de escenarios

En cuanto a la probabilidad de ocurrencia de escenarios, el generador asume por omisión que los escenarios son equiprobables. De lo contrario, el usuario debe especificar la probabilidad de ocurrencia de cada escenario, es decir, la probabilidad de ocurrencia para cada nodo hoja. A partir de estos valores, el generador calcula recurrentemente hacia el nodo raíz la probabilidad de ocurrencia para el resto de los nodos. Otro punto a analizar es la correlación entre los parámetros aleatorios. Dado que en nuestro estudio experimental, al igual que para otros trabajos de la literatura, se utilizan árboles de aridad dos, se asume correlación entre los parámetros aleatorios. En la realidad, es esperable una correlación entre la demanda y los retornos, pero generalmente dicha correlación sucede en distintos períodos temporales, ya que tener una demanda alta en un período, puede implicar tener una alta cantidad de retornos,

pero en períodos futuros. De todas maneras, el generador permite establecer automáticamente árboles de aridad cuatro con parámetros aleatorios independientes.

6.2. Selección de datos y configuración de ejecución

La generación de datos para la mayoría de los parámetros del problema se basa en la estrategia utilizada por Retel Helmrich et al. (2014), en el cual los autores desarrollan un repositorio para problemas con remanufacturación determinista. Algunos parámetros deben ser modificados para atender las particularidades del problema que incorpora incertidumbre. Asimismo, se deben considerar nuevos parámetros propios del problema con esta extensión. A continuación se detalla la generación de parámetros seleccionada en este trabajo:

Aridad. Si bien el generador de datos permite trabajar con árboles de distinta aridad, se optó por realizar las pruebas con árboles completos de aridad dos. Dado que el número de nodos ($|N|$) es exponencial en la cantidad de periodos ($|T|$) para árboles completos: $|N| = 2^{|T|} - 1$, utilizar una mayor aridad no permitiría generar problemas con una cantidad de períodos considerable.

Probabilidades de escenarios. Se asumen escenarios equiprobables. Este enfoque ha sido utilizado anteriormente en la literatura para otros problemas de dimensionado de lote estocástico. De esta manera, la probabilidad de que el estado de la realidad pase de un nodo padre a cualquiera de sus nodos hijos es la misma.

Tamaño de problema. Se realizaron pruebas sobre instancias del problema con un horizonte de planificación de 6, 8, 10 y 12 períodos. Dados los recursos computacionales con los que se cuenta, y el crecimiento exponencial de la cantidad de nodos al aumentar la cantidad de períodos, utilizar tamaños mayores implica un tiempo elevado de generación del problema para la herramienta. El problema de considerar tamaños superiores también se observa en la generación de instancias, ya que los archivos de datos tienen un peso elevado, y tiempo de generación no despreciable a partir de determinados tamaños. Finalmente, un tamaño mayor del problema dificulta la interpretación de las soluciones.

Demanda. Al igual que en Retel Helmrich et al. (2014), en todos los casos se trabaja con una demanda generada según una distribución Normal de media 100 y desvío estándar 50 ($\mu_d = 100$; $\sigma_d = 50$).

Retornos. Los valores de los retornos están también basados en Retel Helmrich et al. (2014). Se generan a partir de variables aleatorias distribuidas en una Normal con tres

combinaciones de parámetros: casos con cantidad de retornos bajo ($\mu_r = 10$; $\sigma_r = 5$), casos con cantidad de retornos medio ($\mu_r = 50$; $\sigma_r = 25$), casos con cantidad de retornos alto ($\mu_r = 90$; $\sigma_r = 45$).

Costo de inventario. En todos los casos se trabaja con costos de inventario de valor 1 tomados de las pruebas realizadas por Retel Helmrich et al. (2014). Los autores indican que la variación de estos costos en el tiempo no tiene un impacto importante en la dificultad de resolución del problema.

Costo de configuración. Los valores para los costos de configuración son tomados de las pruebas realizadas por Retel Helmrich et al. (2014), donde se consideran los mismos valores para la producción de artículos nuevos como remanufacturados, siendo estos 125, 250 y 500. Asimismo, se realizaron pruebas preliminares con valores menores a 125. Las soluciones obtenidas en estas pruebas son poco realistas, y se observa que se activa la configuración de producción en todos o casi todos los nodos. Análogamente, se realizaron pruebas para valores mayores a 500, donde se observa que en la mayoría de los nodos no se activa la producción. Por este motivo, se decidió no experimentar con valores fuera de estos rangos. El comportamiento observado para estas dos situaciones, está basado en el concepto de tiempo entre órdenes (TBO, Time Between Orders), que analiza el ratio entre los costos de configuración y los costos de inventario en el problema clásico sin retornos. Este cálculo brinda una idea aproximada de cada cuantos períodos se activa la producción, pudiendo determinar la forma de la solución de algunos problemas de producción.

El solver seleccionado para resolver las instancias es CPLEX 12.8. con la herramienta AMPL, sobre el sistema operativo LINUX con versión de kernel 3.10.0-1062.1.1.el7.x86_64. El procesador de la computadora es de 3.00GHz, Intel(R) Core(TM) i7-5960X y se dispone de 64GB RAM.

Las posibles estrategias de asignación de valores a los parámetros del problema descripta anteriormente, permiten generar un total de 36 casos diferentes, dado que se tienen 4 tamaños de problema, 3 valores de costo de configuración, y 3 casos de retornos. Para cada caso, se generan cinco instancias. Para cada formulación a estudiar, se ejecutan un total de diez pruebas, de las cuales cinco corresponden a la ejecución sin utilizar cortes del solver, y cinco a la ejecución de esas instancias incorporando cortes del solver. La cantidad y tipo de cortes a incorporar, es la configurada por defecto por el solver. Se considera que una instancia es resuelta a optimalidad cuando la diferencia entre el valor objetivo de la mejor solución factible y la mejor cota inferior encontrada es menor al épsilon del procesador. Se le asigna como un criterio de parada un tiempo límite de ejecución al solver de 600s para cada

instancia de la formulación que no puede resolver a optimalidad.

En las siguientes secciones se estudia el desempeño de las formulaciones presentadas en las Secciones 4.1 y 4.2 para las instancias generadas de acuerdo a la descripción anterior. Los casos a analizar se agrupan acorde al tamaño del problema, es decir la cantidad de períodos en el horizonte de planificación, ya que la dificultad de resolución del mismo depende en gran medida de la cantidad de variables binarias de la instancia. Mientras para problemas de tamaño pequeño en general se puede esperar obtener soluciones óptimas, para problemas de tamaño mayor existen instancias para las que ni siquiera es posible encontrar soluciones factibles en el tiempo asignado para alguna de las formulaciones. De esta manera, para cada tamaño de problema se estudian nueve casos diferentes, resultado de considerar los 3 valores de costo de configuración y 3 casos de retornos. El análisis para los casos con tamaño de problema pequeño, mediano y grande se presentan en las Secciones 6.3, 6.4 y 6.5 respectivamente.

El desempeño de la heurística es estudiado para instancias del problema de tamaño mediano y grande. Como fue mencionado, para problemas de tamaño pequeño se espera que las estrategias de resolución exacta devuelvan soluciones óptimas en tiempos de ejecución razonables. Por lo tanto, para esos casos el estudio se centra en analizar el desempeño de las distintas formulaciones presentadas en los Capítulos 3 y 4. Para la heurística, el análisis para casos de tamaño mediano se presenta en la Sección 6.4 y para tamaño grande en la Sección 6.5.

En las siguientes secciones se utiliza la siguiente nomenclatura para referirse a las distintas estrategias de resolución:

- **Natural:** formulación Natural sin cortes de solver de la Sección 3.3.1
- **IV:** formulación con el agregado de inecuaciones válidas derivadas en las Proposiciones 8 y 9, sin cortes de solver, de la Sección 4.2.1
- **FL:** formulación de localización de instalación sin cortes de solver de la Sección 4.1.1
- **Natural-cortes:** formulación Natural con cortes de solver
- **IV-cortes:** formulación IV con cortes de solver
- **FL-cortes:** formulación FL con cortes de solver
- **R-and-F:** resolución con heurística relax-and-fix de la Sección 5.4

6.3. Estudio para instancias de tamaño pequeño

En esta sección se estudia el desempeño de las formulaciones presentadas para problemas de tamaño pequeño. Se realizan pruebas sobre instancias del problema con horizonte de planificación de 6 y 8 períodos. El resto de los parámetros se seleccionan como fue detallado en la Sección 6.2. Para cada valor de costo de configuración, se estudian casos con distintas distribuciones de cantidad de retornos: bajo, medio y alto. Dado que se consideran árboles completos de aridad 2, para las instancias con horizonte de tamaño 6, se tiene un total de 32 escenarios, 63 nodos y 126 variables binarias (ver cálculo de cantidad de nodos de la Sección 3.2). Para las instancias con horizonte de tamaño 8, se tiene un total de 128 escenarios, 225 nodos y 450 variables binarias. En las Tablas 6.1, 6.2 y 6.3 se presentan resultados promedio de las corridas para las instancias de tamaño 6, con costos de configuración de 125, 250 y 500 respectivamente. En cada uno de las tablas se presentan los resultados para tres casos, correspondientes a las tres distintas distribuciones de retornos explicada en 6.2. Análogamente, los resultados para las instancias de tamaño 8, con costos de configuración de 125, 250 y 500 se presentan en las Tablas 6.4, 6.5 y 6.6. Para cada caso de prueba, se reportan los siguientes resultados:

LP-gap: Para la formulación Natural y las reformulaciones, se reporta la brecha de relajación a programación lineal, con el objetivo de comparar las soluciones óptimas al problema relajado de las formulaciones. Para cada formulación, esta brecha se calcula como la diferencia relativa entre la solución óptima al problema relajado de la formulación, y el valor objetivo de la mejor solución factible encontrada por cualquier formulación. La brecha se calcula para cada instancia y cada formulación, y luego se reporta el valor del promedio entre todas las instancias corridas para cada formulación. Dada una formulación f , el cálculo LP-gap para f es:

$$\text{LP-gap}_f = \frac{VMS - Z_f^{PL}}{VMS},$$

donde VMS : valor mejor solución entre todas las formulaciones, y Z_f^{PL} : valor de relajación a programación lineal de f .

MIP-gap: Se reporta la brecha (gap) de optimalidad, calculado para cada formulación como la diferencia relativa entre el valor objetivo de la solución factible encontrada por el solver para la formulación, y la mejor cota inferior encontrada por el solver tras la ejecución. En caso de que una instancia se pueda resolver a optimalidad, el MIP-gap es 0. Este valor se

calcula para cada instancia y cada formulación, y luego se reporta el valor del promedio entre todas las instancias corridas para cada formulación. Dada una formulación f , el cálculo MIP-gap para f es:

$$\text{MIP-gap}_f = \frac{Z_f^{fact} - Z_f^{inf}}{Z_f^{fact}},$$

donde Z_f^{fact} : valor mejor solución factible conocida de f , y Z_f^{inf} : valor cota inferior de f reportada por el solver tras la ejecución.

Tiempo de ejecución: Para la formulación Natural y las reformulaciones, el tiempo de ejecución coincide con el asignado en el criterio de parada en la Sección 6.2, salvo en los casos en los que se obtenga una solución óptima en tiempo menor.

Porcentaje de soluciones óptimas Para cada caso y cada formulación, se reporta el porcentaje de instancias que pudo ser resuelto a optimalidad en un tiempo menor al asignado en el criterio de parada en la Sección 6.2.

Tabla 6.1 Pruebas con 6 períodos y costo de configuración 125

Retornos		Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	68,24	6,34	6,34	68,24	6,34	6,34
	mipgap (%)	0,00	0,00	0,00	0,00	0,00	0,00
	tiempo (s)	16,58	1,13	1,75	0,26	0,44	0,94
	óptimas (%)	100	100	100	100	100	100
medio	lpgap (%)	70,97	10,48	10,40	70,97	10,48	10,40
	mipgap (%)	4,84	0,00	0,00	0,00	0,00	0,00
	tiempo (s)	602,15	14,17	18,80	1,34	0,94	4,88
	óptimas (%)	0	100	100	100	100	100
alto	lpgap (%)	55,63	11,70	11,60	55,63	11,70	11,60
	mipgap (%)	0,22	0,00	0,00	0,00	0,00	0,00
	tiempo (s)	496,38	9,12	12,53	0,61	0,77	2,62
	óptimas (%)	60	100	100	100	100	100

6.3.1. Interpretación de resultados

Para ambos tamaños de problema, se observa que para cada costo de configuración, las instancias son más fáciles de resolver cuando la cantidad de retornos es baja. Una posible

Tabla 6.2 Pruebas con 6 períodos y costo de configuración 250

Retornos		Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	61,02	5,60	5,56	61,02	5,60	5,56
	mipgap (%)	0,00	0,00	0,00	0,00	0,00	0,00
	tiempo (s)	2,75	3,23	2,23	1,16	1,65	0,90
	óptimas (%)	100	100	100	100	100	100
medio	lpgap (%)	64,74	9,34	9,20	64,74	9,34	9,20
	mipgap (%)	2,95	0,09	0,20	0,00	0,00	0,00
	tiempo (s)	601,86	190,82	127,45	2,81	4,91	18,04
	óptimas (%)	0	80	80	100	100	100
alto	lpgap (%)	59,27	12,89	12,60	59,27	12,89	12,60
	mipgap (%)	0,91	0,24	0,21	0,00	0,00	0,00
	tiempo (s)	558,28	258,24	262,37	2,40	3,78	24,68
	óptimas (%)	20	80	80	100	100	100

Tabla 6.3 Pruebas con 6 períodos y costo de configuración 500

Retornos		Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	54,21	7,34	7,31	54,21	7,34	7,31
	mipgap (%)	0,00	0,00	0,00	0,00	0,00	0,00
	tiempo (s)	0,50	0,43	0,81	0,21	0,54	0,72
	óptimas (%)	100	100	100	100	100	100
medio	lpgap (%)	59,15	8,62	8,60	59,15	8,62	8,60
	mipgap (%)	0,00	0,00	0,02	0,00	0,00	0,00
	tiempo (s)	53,35	29,12	37,99	1,03	2,09	7,40
	óptimas (%)	100	100	100	100	100	100
alto	lpgap (%)	59,25	11,25	11,00	59,25	11,25	11,00
	mipgap (%)	0,18	0,06	0,40	0,00	0,00	0,00
	tiempo (s)	246,84	182,91	157,19	4,66	6,34	17,45
	óptimas (%)	80	80	80	100	100	100

Tabla 6.4 Pruebas con 8 períodos y costo de configuración 125

Retornos		Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	75,05	6,16	6,15	75,05	6,16	6,15
	mipgap (%)	13,64	0,94	1,18	0,00	0,00	0,00
	tiempo (s)	612,01	602,47	602,51	16,08	15,36	171,01
	óptimas (%)	0	0	0	100	100	100
medio	lpgap (%)	73,24	10,13	9,96	73,24	10,13	9,96
	mipgap (%)	24,58	2,95	3,04	0,26	0,05	1,24
	tiempo (s)	613,36	602,89	602,27	504,67	258,20	600,17
	óptimas (%)	0	0	0	40	80	0
alto	lpgap (%)	56,76	11,34	11,29	56,76	11,34	11,29
	mipgap (%)	14,82	3,24	3,25	0,13	0,00	0,68
	tiempo (s)	617,13	603,34	602,89	442,73	529,28	600,14
	óptimas (%)	0	0	0	40	100	0

Tabla 6.5 Pruebas con 8 períodos y costo de configuración 250

Retornos		Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	69,86	6,51	6,51	69,86	6,51	6,51
	mipgap (%)	7,61	1,75	1,90	0,02	0,00	0,24
	tiempo (s)	613,98	602,64	602,38	156,70	121,53	591,80
	óptimas (%)	0	0	0	80	100	20
medio	lpgap (%)	71,40	9,83	9,49	71,40	9,83	9,49
	mipgap (%)	20,70	3,65	3,56	1,51	0,71	2,83
	tiempo (s)	612,68	602,41	602,27	601,14	601,33	600,35
	óptimas (%)	0	0	0	0	0	0
alto	lpgap (%)	62,90	12,49	12,23	62,90	12,49	12,23
	mipgap (%)	14,62	3,40	3,52	1,01	0,18	1,85
	tiempo (s)	616,21	602,78	602,14	603,88	408,74	600,13
	óptimas (%)	0	0	0	0	80	0

Tabla 6.6 Pruebas con 8 períodos y costo de configuración 500

Retornos		Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	63,10	6,40	6,39	63,10	6,40	6,39
	mipgap (%)	3,98	1,96	2,06	0,03	0,00	0,43
	tiempo (s)	613,13	602,42	603,12	243,71	179,41	600,15
	óptimas (%)	0	0	0	80	100	0
medio	lpgap (%)	66,69	9,96	9,76	66,69	9,96	9,76
	mipgap (%)	13,69	4,13	4,09	2,01	2,46	4,33
	tiempo (s)	614,67	602,45	600,36	600,38	600,42	600,14
	óptimas (%)	0	0	0	0	0	0
alto	lpgap (%)	60,76	10,59	10,25	60,76	10,59	10,25
	mipgap (%)	10,29	3,69	3,70	1,43	0,26	2,33
	tiempo (s)	620,42	602,54	602,65	603,32	374,96	600,14
	óptimas (%)	0	0	0	0	80	0

explicación para este comportamiento es que las instancias con pocos retornos guardan una mayor similitud con el problema dimensionado de lotes sin opciones de remanufactura. Como fue visto en Ahmed et al. (2003), algunas estrategias, como las reformulaciones de localización de instalación, devuelven buenas soluciones para estos problemas. Asimismo, la implementación del solver puede facilitar la resolución de instancias de dicho problema. La mayor dificultad de resolver casos con cantidad de retornos medios y altos también se puede explicar considerando que, debido al aumento en la cantidad de retornos, existe una mayor posibilidad de alternar entre producción y remanufactura. Los tiempos de ejecución para la formulación Natural sin cortes de solver fueron los más elevados en todos los casos. En algunos de estos casos, la diferencia fue de varios órdenes de magnitud mayor al comparar con cualquiera de las otras estrategias. En los casos en los que ninguna estrategia pudo obtener soluciones óptimas, esta formulación además tuvo las brechas de optimalidad más grandes. Al comparar los tiempos de ejecución con cortes de solver, en general se observa menores tiempos para IV-cortes, que FL-cortes. En relación a la brecha de relajación a programación lineal de IV y FL, se observa que la misma es muy similar para todos los casos. Dado que el cálculo de esta brecha se realiza con el valor objetivo de la mejor solución factible encontrada entre todas las formulaciones, se concluye que las relajaciones de ambos problemas son similares. Este resultado es similar al obtenido por los autores que estudiaron las versiones deterministas del problema con remanufactura. Ambas relajaciones son además superiores a la de Natural. Se corrobora que con instancias de tamaño pequeño, tener relajaciones más próximas al caso convexo facilita al solver encontrar la solución óptima. Por este motivo, y

como se verá a continuación para ambos tamaños de problema, cuando no se utilizan cortes de solver, el desempeño de las reformulaciones tiende a ser superior a la formulación Natural.

6.3.2. Análisis para seis períodos

Para el caso de seis períodos, las reformulaciones encontraron una cantidad elevada de soluciones óptimas, de al menos el 80% de las instancias analizadas en cada caso, con y sin cortes del solver. La formulación Natural tuvo un desempeño que varió dependiendo tanto del costo de configuración como la cantidad de retornos. Los peores resultados se obtuvieron con un costo de configuración de 250, y cantidad de retornos medios y altos.

Comparando las formulaciones sin cortes de solver, en siete de los casos IV tuvo el mejor desempeño, en tiempo de ejecución, porcentaje de soluciones óptimas, o en gap de optimalidad. En los dos casos restantes, el mejor desempeño fue el de FL. En todos los casos, Natural tuvo un peor desempeño que alguna de las dos reformulaciones. Al utilizar cortes de solver, para costos de configuración de 125 el tiempo de ejecución fue siempre menor a tres segundos para cualquier formulación. Para los otros valores de costos, el desempeño de IV y Natural fue similar, y con un tiempo de ejecución ligeramente menor al de FL en la mayoría de los casos.

Al comparar el beneficio en la utilización o no de cortes de solver, en general se obtienen mejores tiempos de resolución incorporando cortes para los casos particulares donde se obtienen soluciones óptimas. Esta situación se observa especialmente en la formulación Natural, cuyo desempeño se acerca al de las reformulaciones al incorporar estos cortes. Para las reformulaciones, las diferencias son menores. De todas maneras, en tres casos fue posible obtener la solución óptima de una instancia de IV que no se obtuvo al no utilizarlos.

6.3.3. Análisis para ocho períodos

Para el caso de ocho períodos, costo de configuración 125, 250 y retornos bajos, tanto la formulación Natural como las reformulaciones encontraron soluciones óptimas. Para el resto de los casos de ocho períodos, las únicas formulaciones que encontraron soluciones óptimas fueron Natural y IV, ambas con cortes del solver. El desempeño de IV fue en general superior, ya que todas las instancias en las que se llegó al óptimo con Natural, se pudieron resolver también con IV, salvo para el caso con retornos de 125, y retornos altos. La mayor diferencia se observa para el caso con costos de configuración 500, y retornos altos, en el que el 80% de las instancias se pudieron resolver con IV, mientras que ninguna otra reformulación pudo

resolver una instancia.

Las instancias sin cortes de solver no encontraron soluciones óptimas. Las brechas de optimalidad fueron siempre peores para la formulación Natural. En siete de los nueve casos, IV tuvo los menores valores de gap de optimalidad. Sin embargo, al igual que para las instancias de tamaño 6, estos valores fueron similares a los de FL.

Al utilizar cortes de solver, la diferencia en el desempeño de las reformulaciones aumenta. En estos casos, tanto las brechas de optimalidad, los tiempos de ejecución como el porcentaje de soluciones óptimas fueron considerablemente mejores para IV. Una posible explicación, es la incorporación de cortes de flujo generados por el solver. Dado que la idea para su generación es representar el problema como una red de flujo, estos pueden guardar similitud con la formulación FL, y por lo tanto ser redundantes y empeorar el rendimiento del solver. Al comparar IV y Natural, el rendimiento de IV fue superior, observándose la mayor diferencia para los casos con retornos altos, independientemente del costo de configuración.

Se detectaron ventajas importantes en la utilización de los cortes del solver para las instancias con retornos bajos. Como fue mencionado anteriormente, cuando la cantidad de retornos es baja, el problema se asemeja a un problema de dimensionado de lotes sin remanufactura. Por otro lado, también se obtuvieron ventajas en la utilización de los cortes del solver sobre la formulación IV para casos de retornos altos.

6.4. Estudio para instancias de tamaño mediano

En esta sección se estudia el desempeño de las distintas formulaciones presentadas y la heurística para problemas de tamaño mediano, con instancias del problema cuyo horizonte de planificación es de 10 períodos. Al igual que para las instancias de tamaño pequeño, el resto de los parámetros se seleccionan acorde a lo detallado en la Sección 6.2. Para cada valor de costo de configuración, se estudian casos acorde a las distribuciones de cantidad de retornos: bajo, medio y alto. Las instancias de estos casos de prueba tienen un total de 512 escenarios, 1023 nodos y 2046 variables binarias.

Para instancias de tamaño mediano se incorpora el análisis del desempeño de la heurística. Al comparar la heurística, nos interesa particularmente analizar el valor objetivo de la mejor solución factible obtenida y el tiempo de ejecución. No es posible garantizar que las cotas inferiores obtenidas en las distintas etapas de ejecución de la heurística sean cotas para el problema original, ya que en las etapas se resuelven problemas más restringidos. Por lo tanto, se omiten esos valores en las tablas de resultados. En las Tablas 6.7, 6.8 y 6.9 se presentan resultados promedio de las corridas para las instancias con costos de configuración de 125, 250 y 500 respectivamente. En cada una de las tablas se presentan los resultados para los tres casos correspondientes a las distintas distribuciones de retornos definidas en la Sección 6.2. Para cada caso de prueba, se reportan los siguientes resultados:

LP-gap: Para la formulación Natural y las reformulaciones, se reporta la brecha de relajación a programación lineal, calculada como la diferencia relativa entre el valor objetivo de la mejor solución factible encontrada entre todas las formulaciones, y el valor objetivo de la solución óptima al problema relajado de la formulación. El cálculo de este valor a reportar se puede ver en la Sección 6.3.

MIP-gap: Para la formulación Natural y las reformulaciones, se reporta la brecha (gap) de optimalidad, calculada para cada formulación como la diferencia relativa entre el valor objetivo de la mejor solución factible encontrada, y la cota inferior encontrada luego de la ejecución de dicha formulación. El cálculo de este valor es análogo al de la Sección 6.3.

Tiempo de ejecución: La ejecución de la formulación Natural y las reformulaciones termina en el caso de encontrarse una solución óptima en un tiempo de ejecución menor al asignado en el criterio de parada en la Sección 6.2. La heurística, por otro lado, podrá terminar en un tiempo menor al criterio de parada sin necesariamente haber encontrado una solución óptima para el problema original.

Valor de la función objetivo: Se incluye en el análisis el valor de la función objetivo de la mejor solución encontrada, tanto por la heurística como por las reformulaciones.

A partir de este tamaño de instancia, no fue posible obtener soluciones óptimas. De todas maneras, la brecha de optimalidad fue menor a 14% en promedio para las reformulaciones en todos los casos. Análogamente a lo comentado para las instancias de tamaño pequeño, la brecha de relajación a programación lineal de IV y FL son muy similares. Sin embargo, para este tamaño de instancia, se observa una mayor diferencia en la brecha de Natural con respecto a las anteriores. El principal motivo que explica esta observación, es que las cotas inferiores obtenidas a través de la relajación a programación lineal son considerablemente superiores en las reformulaciones. Se corrobora que la incorporación de nuevas restricciones en ambas reformulaciones permite tener relajaciones más próximas al casco convexo. Las brechas de optimalidad fueron mayores para los casos con retornos medios, mostrando que aparentemente estos casos son más difíciles de resolver. Esto se puede deber a que en estos casos puede existir una mayor posibilidad de alternar entre producción y remanufacturación.

Cuando no se consideran cortes del solver, la brecha de optimalidad de IV y FL ha sido similar. A pesar de que FL tiene una relajación a programación lineal cuya solución es una mejor cota que IV, en algunos casos la brecha de optimalidad fue ligeramente menor para IV. La explicación de la diferencia en la brecha, se obtiene de observar los valores de función objetivos obtenidas con ambas formulaciones. Por otro lado, al igual que para tamaño pequeño, se vuelve a observar una peor brecha en Natural con respecto a las reformulaciones. De hecho, esta brecha se acrecentó en Natural para instancias de tamaño medio. Este resultado se puede deber a dos motivos. Por un lado, como fue mencionado al principio de esta sección las cotas inferiores de las reformulaciones son superiores a las de Natural. Por otro lado, los valores de función objetivo para Natural fueron peores en ocho de los casos a los de alguna de las dos reformulaciones. Sin embargo, se debe notar que hubo un caso en los que Natural obtuvo el mejor valor de función objetivo. Este caso fue el de retornos medios, con costo de configuración de 500. En todos los casos las reformulaciones permiten obtener mejores cotas del valor óptimo.

Al considerar cortes del solver, se observa una mejoría en las brechas de optimalidad de IV y Natural. Esta mejoría no se observa para FL, incluso obteniéndose peores brechas al agregar cortes en alguno de los casos. Una posible explicación es que el solver incorpora cortes de flujo, que pueden guardar similitud con FL que es una representación de una red de flujo. Para

Tabla 6.7 Pruebas con 10 períodos y costo de configuración 125

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes	R-and-F	
bajo	lpgap (%)	78,54	5,70	5,67	78,54	5,70	5,67	no aplica
	mipgap (%)	27,09	2,95	3,11	0,72	0,27	2,48	no aplica
	tiempo (s)	604,76	600,28	600,68	600,37	601,35	600,47	9,86
	objetivo	1410,33	1409,40	1409,87	1409,49	1409,94	1415,03	1443,17
medio	lpgap (%)	77,14	11,75	11,54	77,14	11,75	11,54	no aplica
	mipgap (%)	37,31	6,88	6,66	2,61	1,92	8,10	no aplica
	tiempo (s)	605,16	600,35	600,91	600,13	600,87	600,56	225,69
	objetivo	1671,67	1673,93	1667,76	1666,01	1665,03	1716,83	1685,74
alto	lpgap (%)	54,13	11,73	11,71	54,13	11,73	11,71	no aplica
	mipgap (%)	20,85	5,15	5,12	1,57	0,99	4,51	no aplica
	tiempo (s)	605,79	600,26	600,82	600,18	601,91	600,53	28,98
	objetivo	1.966,44	1.960,94	1.961,29	1.962,01	1.963,61	1.984,50	1.974,83

Tabla 6.8 Pruebas con 10 períodos y costo de configuración 250

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes	R-and-F	
bajo	lpgap (%)	74,38	6,91	6,91	74,38	6,91	6,91	no aplica
	mipgap (%)	21,21	4,39	4,30	1,81	2,48	5,56	no aplica
	tiempo (s)	604,69	601,72	600,60	600,13	600,30	600,43	9,20
	objetivo	2279,02	2275,08	2274,34	2273,53	2291,35	2315,77	2361,49
medio	lpgap (%)	75,17	12,91	12,20	75,17	12,91	12,20	no aplica
	mipgap (%)	33,04	9,52	9,55	6,87	9,44	14,00	no aplica
	tiempo (s)	605,26	600,71	600,59	600,73	601,17	600,37	135,60
	objetivo	2659,15	2641,26	2651,19	2642,36	2736,30	2818,66	2666,21
alto	lpgap (%)	65,17	13,36	12,99	65,17	13,36	12,99	no aplica
	mipgap (%)	24,75	7,53	6,73	4,72	8,24	9,68	no aplica
	tiempo (s)	606,19	601,38	600,49	600,14	600,76	600,36	18,72
	objetivo	2.873,41	2.876,64	2.866,80	2.875,65	2.986,49	2.973,39	2.885,97

Tabla 6.9 Pruebas con 10 períodos y costo de configuración 500

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes	R-and-F
bajo	lpgap (%)	6,77	6,76	68,27	6,77	6,76	no aplica
	mipgap (%)	4,60	4,34	2,21	3,12	5,51	no aplica
	tiempo (s)	601,38	600,58	600,58	600,26	600,42	5,09
	objetivo	3495,91	3480,67	3477,83	3505,73	3538,72	3650,81
medio	lpgap (%)	11,47	10,99	71,74	11,47	10,99	no aplica
	mipgap (%)	10,67	11,26	8,35	11,57	13,31	no aplica
	tiempo (s)	602,40	602,62	600,13	601,38	600,38	60,31
	objetivo	4215,58	4218,47	4152,22	4345,55	4355,05	4223,55
alto	lpgap (%)	10,77	10,35	63,07	10,77	10,35	no aplica
	mipgap (%)	8,31	6,51	5,72	10,01	10,46	no aplica
	tiempo (s)	602,10	600,46	600,74	600,39	600,31	14,39
	objetivo	4.407,30	4.358,90	4.392,65	4.605,79	4.538,85	4.410,35

costos de 250 y 500, se obtuvieron menores brechas y valores de función objetivo con Natural.

En este tamaño de instancia, se evidencia un deterioro en el valor de función objetivo de las reformulaciones IV y FL con la incorporación de cortes del solver. Para FL, se observa también un deterioro en la brecha de optimalidad. Para Natural, la inclusión de cortes mejora su desempeño en siete de los casos. Las diferencias que se observan al incorporar cortes del solver sobre las distintas formulaciones, se puede deber a que Natural es una formulación con una menor cantidad de restricciones. Las reformulaciones tienen una gran cantidad de restricciones, por lo que la incorporación de nuevas puede deteriorar su funcionamiento. En seis de los nueve casos los mejores valores de función objetivo en promedio se obtienen con las reformulaciones sin utilizar cortes de solver. En los tres restantes, los mejores valores se obtienen con Natural, con cortes de solver.

Con la heurística fue posible obtener valores cercanos a los de las formulaciones en tiempos de ejecución menores. Los valores de función objetivo de la heurística son cercanos especialmente para cantidad de retornos medio y alto para todos los costos de configuración. Esto se debe a que las formulaciones en general dan buenos resultados cuando la cantidad de retornos es baja, por lo que para obtener resultados similares la heurística debería obtener soluciones muy cercanas a la óptima. A excepción del caso con costo 125 y retorno medio, el tiempo de ejecución de la heurística fue menor al 25 % del tiempo utilizado por las formulaciones. Para retornos bajos y altos el tiempo de ejecución fue menor al 5 % del de las formulaciones.

6.5. Estudio para instancias de tamaño grande

En esta sección se estudia el desempeño de las distintas formulaciones presentadas y la heurística para problemas de tamaño grande, con instancias del problema cuyo horizonte de planificación es de 12 períodos. Al igual que para las instancias de tamaño pequeño y mediano, para cada valor de costo de configuración, se estudian casos acorde a las distribuciones de cantidad de retornos: bajo, medio y alto. Dado que se consideran árboles completos de aridad 2, las instancias de estos casos de prueba tienen un total de 2048 escenarios, 4095 nodos y 8190 variables binarias. La ejecución preliminar de instancias de prueba permitieron observar que el desempeño de la reformulación con inecuaciones válidas, y la reformulación de localización de instalaciones se deterioran al considerar horizontes de planificación de 11 o más períodos para algunas instancias. En algunas instancias de estos casos, las reformulaciones mencionadas ni siquiera logran encontrar una solución factible en el tiempo de ejecución considerado. La cantidad de inecuaciones a agregar aumenta considerablemente al aumentar la cantidad de períodos en el horizonte de planificación, ya que la cantidad de nodos del árbol aumenta exponencialmente. Por lo tanto, una explicación posible es que en ambas formulaciones la cantidad de inecuaciones a agregar es tan elevada, que vuelve inmanejable el problema con los recursos con los que se cuenta de tiempo y capacidad de cómputo. Esto no sucede para el caso de la formulación Natural, que tiene una cantidad de restricciones considerablemente menor. Teniendo en consideración estas características, se decidió dividir el análisis en dos partes para estas instancias de tamaño grande. En la Sección 6.5.1, se compara el desempeño de la formulación Natural, con el de las reformulaciones. Como se verá en dicha sección, la formulación Natural no solo obtiene un mayor porcentaje de soluciones factibles, sino que también brinda mejores valores de función objetivo, valor que es necesario para comparar el desempeño de la heurística. Por lo tanto, en la Sección 6.5.2, se compara el desempeño de la heurística, con el de la formulación Natural.

6.5.1. Desempeño de las reformulaciones para instancias de tamaño grande

Como fue mencionado, existen casos en los que un porcentaje considerable de ejecuciones de instancias de las reformulaciones no pueden encontrar soluciones factibles en el tiempo de ejecución asignado. Para esos casos, se muestra el porcentaje de soluciones factibles encontradas sobre el total de ejecuciones realizadas: en la Tabla 6.10 se presenta el porcentaje de soluciones factibles para los casos con costo de configuración de 250, y retornos medios y

altos; en la Tabla 6.11 se presenta el porcentaje de soluciones factibles para los casos con costo de configuración de 500, y retornos medios y altos. Para los restantes cinco casos, todas las reformulaciones encontraron soluciones factibles para todas las instancias. Por lo tanto, el análisis de las reformulaciones para instancias de tamaño grande se centra en estos casos. En las Tablas 6.12, 6.13 y 6.14 se presentan resultados promedio de las corridas. No se muestra el tiempo de ejecución, dado que ni la formulación Natural ni las reformulaciones encontraron soluciones óptimas en el tiempo de ejecución provisto. En la Tabla 6.12 se presentan resultados para instancias con costos de configuración de 125, y distribuciones de retornos bajo, medio y alto. En las Tablas 6.13 y 6.14 se presentan resultados para costos de configuración de 250 y 500 respectivamente y distribuciones de retornos bajo. Para cada caso de prueba, se reportan los siguientes resultados:

LP-gap: Para la formulación Natural y las reformulaciones, se reporta la brecha de relajación a programación lineal, calculada como la diferencia relativa entre el valor objetivo de la mejor solución factible encontrada entre todas las formulaciones, y el valor objetivo de la solución óptima al problema relajado de la formulación. El cálculo de este valor a reportar se puede ver en la Sección 6.3.

MIP-gap: Para la formulación Natural y las reformulaciones, se reporta la brecha (gap) de optimalidad, calculada para cada formulación como la diferencia relativa entre el valor objetivo de la mejor solución factible encontrada, y la mejor cota inferior encontrada luego de la ejecución de dicha formulación. El cálculo de este valor es análogo al de la Sección 6.3.

Valor de la función objetivo: Se incluye el valor de la función objetivo promedio de la mejor solución encontrada por las formulaciones. Este valor permitirá determinar si existe una formulación más conveniente que las demás para comparar el desempeño de la heurística en la siguiente sección. Como ya fue mencionado, no es posible utilizar LP-gap y MIP-gap como medidas de desempeño de la heurística.

Tabla 6.10 Porcentaje de soluciones factibles para 12 períodos y costo de configuración 250

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
medio	100	60	80	100	60	80
alto	100	100	80	100	100	60

A pesar de haber casos con instancias de tamaño grande en los que no es posible obtener soluciones factibles, para los otros casos la brecha de optimalidad en promedio fue siempre menor a 21 % para las reformulaciones con y sin cortes de solver. Al igual que para otros tamaños de instancias, la brecha de relajación a programación lineal de IV y FL son similares.

Tabla 6.11 Porcentaje de soluciones factibles para 12 períodos y costo de configuración 500

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
medio	100	0	60	100	0	40
alto	100	80	80	100	80	80

A diferencia de las instancias de tamaño medio, cuando no se consideran cortes del solver en la mayoría de los casos se observan valores de brechas de optimalidad considerablemente menores en IV que los obtenidos en FL, como se puede ver en las Tablas 6.12, 6.13 y 6.14. Este último resultado es interesante, ya que permite notar que aún con relajaciones a programación lineal muy similares, el desempeño de una reformulación puede ser superior al de otra. A modo de ejemplo, en la Tabla 6.14, se observa el caso con retornos bajos en el que la diferencia de $LP - gap$ entre las reformulaciones fue menor a 0,01 %, mientras que la brecha $MIP - gap$ fue de 13,48 % y 20,45 % para VI y FL respectivamente.

Por otro lado, las brechas de relajación a programación lineal y de optimalidad de las reformulaciones fueron menores que la de la formulación Natural, mientras que los valores de función objetivo para Natural fueron mejores en los cinco casos. Se observa que mientras las reformulaciones permiten obtener buenas cotas inferiores, la obtención de estas cotas pueden perjudicar la obtención de mejores soluciones factibles. De todos modos, las reformulaciones mantuvieron valores de función objetivo ligeramente mayores a los de Natural, con brechas de optimalidad menores. Dadas las ventajas que se tienen con la formulación Natural para obtener mayor porcentaje de soluciones factibles y buenos valores objetivos, en la siguiente sección se utiliza dicha formulación para comparar con la heurística.

La utilización de los cortes sobre las tres formulaciones, parece no mejorar la solución factible encontrada a partir de este tamaño del problema. Al considerar cortes del solver, en general se observa un empeoramiento de las soluciones encontradas de todas las formulaciones. Para Natural, en todos menos un caso las soluciones fueron mejores al no utilizar los cortes. Para IV y FL, las soluciones fueron en general mejores al no utilizar cortes de solver. Este resultado contrasta con lo observado para instancias de tamaño pequeño, y con lo que sucede con Natural para tamaño mediano. Una posible explicación de estas observaciones, es que la cantidad de restricciones de las formulaciones crece exponencialmente con el tamaño del problema. A partir de este tamaño, la cantidad de restricciones puede ser alta, de modo que la incorporación de nuevas no facilita la resolución por parte del solver. Por otro lado,

Tabla 6.12 Pruebas con 12 periodos y costo de configuración 125

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
bajo	lpgap (%)	81,60	6,97	6,92	81,60	6,97
	mipgap (%)	40,73	6,71	7,39	3,10	2,98
	objetivo	1739,12	1744,15	1756,95	1727,93	1740,61
medio	lpgap (%)	77,97	13,05	12,75	77,97	13,05
	mipgap (%)	48,71	20,05	19,34	20,97	17,27
	objetivo	1972,60	2180,09	2153,52	2313,34	2209,63
alto	lpgap (%)	58,62	13,81	13,77	58,62	13,81
	mipgap (%)	28,43	11,43	14,91	7,03	7,67
	objetivo	2319,52	2357,14	2439,20	2377,53	2408,40
						2464,96

Tabla 6.13 Pruebas con 12 períodos y costo de configuración 250

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
lpgap (%)	77,48	8,96	8,94	77,48	8,96	8,94
mipgap (%)	34,13	10,43	12,20	7,74	8,78	11,79
objetivo	2730,27	2792,35	2845,24	2781,10	2803,58	2868,62

Tabla 6.14 Pruebas con 12 períodos y costo de configuración 500

Retornos	Natural	IV	FL	Natural-cortes	IV-cortes	FL-cortes
lpgap (%)	72,57	9,04	9,03	72,57	9,04	9,03
mipgap (%)	25,97	13,48	20,45	12,27	11,74	19,84
objetivo	4197,35	4433,85	4832,38	4458,17	4423,75	4832,33

en algunos casos, las brechas mejoraron ligeramente con la utilización de cortes del solver, debido a un mejoramiento de las cotas inferiores halladas.

6.5.2. Desempeño de la heurística para instancias de tamaño grande

Al comparar la heurística, nos interesa particularmente analizar la mejor solución factible obtenida y el tiempo de ejecución. No es posible garantizar que las cotas inferiores obtenidas en las distintas etapas de ejecución de la heurística sean cotas para el problema original, ya que en las etapas se resuelven problemas más restringidos. Por lo tanto, se omiten esos valores en las tablas de resultados.

Al igual que para otros tamaños menores de problemas, las brechas de optimalidad de la formulación Natural son sensiblemente mayores a los de las reformulaciones. Para el caso de instancias de tamaño grande se observa que, pese a esto, las soluciones óptimas obtenidas con la formulación Natural son muy cercanas a las obtenidas con las reformulaciones, y para algunas instancias mejores. Es decir, si bien con la formulación Natural es más difícil obtener para el solver la optimalidad de una solución, las soluciones parecen ser de buena calidad. Esto dificulta el análisis del desempeño de la heurística, ya que las soluciones obtenidas por la formulación Natural pueden ser cercanas a la óptima. Por lo tanto, aún en los casos en los que la heurística se encuentre cerca del óptimo, si se compara únicamente la diferencia relativa de las mejores soluciones obtenidas por ambas estrategias, esta diferencia será pequeña. De modo de poder visualizar más claramente si la heurística tiene la capacidad de mejorar las soluciones, se incorpora a las tablas de resultados de esta sección los cálculos de *Gap vs blb* y *Mejora Relax-and-Fix*, que permiten estudiar en que porcentaje la heurística logra cerrar las brechas de optimalidad, y se explicarán a continuación. Con estos cálculos, es posible estudiar el desempeño de la heurística, aún cuando las soluciones obtenidas por el solver son cercanas a la óptima. Una observación importante sobre las reformulaciones, que se desprende de los estudios presentados en este informe donde se incluye el *LP-gap*, es que devuelven buenas cotas inferiores. Esto es cierto incluso para instancias en las que no se puede encontrar soluciones factibles. Por este motivo para calcular una mejor brecha de optimalidad que la de la formulación Natural, en el cálculo *Gap vs blb* se utiliza la mejor cota inferior obtenida por cualquiera de las reformulaciones a través de la resolución del problema relajado a programación lineal. Para cada caso de prueba, se reportan los siguientes resultados:

Gap vs blb: Esta brecha de optimalidad se calcula como la diferencia relativa entre la mejor solución encontrada por la estrategia de resolución que se está estudiando (heurística,

formulación Natural), y la mejor cota inferior encontrada por cualquiera de las formulaciones. Para esta cota inferior se considera también, entonces, la formulación con inecuaciones válidas, y localización de instalación. Este cálculo permite tener una idea de la cercanía de la solución de la heurística con respecto al óptimo. Asimismo, permite comparar las mejores soluciones encontradas por ambas estrategias. El cálculo se realiza para cada instancia y cada estrategia de resolución, y luego se reporta el valor del promedio entre todas las instancias corridas para cada estrategia. Dada una estrategia de resolución e , que puede ser tanto una formulación como la heurística. El cálculo Gapvsblb_e para e es:

$$\text{Gapvsblb}_e = \frac{Z_e^{fact} - blb}{Z_e^{fact}},$$

donde Z_e^{fact} : valor mejor solución factible de la estrategia de resolución e , y blb : mejor cota inferior, considerando la relajación a programación lineal de las formulaciones Natural, IV y FL.

Mejora Relax-and-Fix: Se calcula como la diferencia relativa de las brechas de optimalidad calculadas en el punto anterior, entre la heurística y la estrategia de resolución exacta. Este valor permite conocer el porcentaje de la brecha obtenida con la formulación Natural que fue reducido al utilizar la heurística. De esta manera, se puede analizar el desempeño de la heurística, aún cuando las soluciones obtenidas por el solver son cercanas a la óptima. El valor se calcula para cada instancia y la formulación Natural con y sin cortes del solver, y luego se reporta el valor del promedio entre todas las instancias corridas para cada formulación. Dada una formulación f , a la que se le pueden agregar o no cortes de solver, y dada una heurística h , el cálculo LP-gap para f es

$$\text{MejoraRelax-and-Fix}_f = \frac{\text{Gapvsblb}_f - \text{Gapvsblb}_h}{\text{Gapvsblb}_f},$$

donde Gapvsblb_f : es la brecha calculada en el punto anterior aplicada a la formulación f , y Gapvsblb_h : es la brecha calculada en el punto anterior aplicada a la heurística.

Tiempo de ejecución: La ejecución de la formulación Natural termina en el caso de encontrarse una solución óptima en un tiempo de ejecución menor al asignado en el criterio de parada en la Sección 6.2. La heurística, por otro lado, podrá terminar en un tiempo menor al

criterio de parada sin necesariamente haber encontrado una solución óptima para el problema original.

En las Tablas 6.15, 6.16 y 6.17 se presentan resultados promedio de las corridas para las instancias con costos de configuración de 125, 250 y 500 respectivamente. En cada uno de las tablas se presentan los resultados para los tres casos correspondientes a las distintas distribuciones de retornos explicada en la Sección 6.2.

Tabla 6.15 Comparación desempeño de la heurística con formulación natural, con 12 períodos y costo de configuración 125

Retornos		Natural	Natural-cortes	R-and-F
bajo	gap vs blb (%)	3	2	3
	tiempo (s)	601,83	600,20	570,82
	mejora RnF (%)	-7	-30	no aplica
medio	gap vs blb (%)	7	21	6
	tiempo (s)	601,12	600,24	610,36
	mejora RnF (%)	7	68	no aplica
alto	gap vs blb (%)	4	6	3
	tiempo (s)	602,03	600,94	499,66
	mejora RnF (%)	15	47	no aplica

Tabla 6.16 Comparación desempeño de la heurística con formulación natural, con 12 períodos y costo de configuración 250

Retornos		Natural	Natural-cortes	R-and-F
bajo	gap vs blb (%)	6	8	6
	tiempo (s)	601,60	600,23	296,72
	mejora RnF (%)	-8	12	no aplica
medio	gap vs blb (%)	13	23	10
	tiempo (s)	601,29	600,32	600,57
	mejora RnF (%)	21	56	no aplica
alto	gap vs blb (%)	0,10	0,16	0,08
	tiempo (s)	601,32	600,21	458,75
	mejora RnF (%)	21	47	no aplica

La heurística tuvo un desempeño superior a Natural, con y sin cortes de solver, para todos los casos con retornos medios y altos. Este resultado se evidencia con el valor de Mejora

Tabla 6.17 Comparación desempeño de la heurística con formulación natural, con 12 períodos y costo de configuración 500

Retornos		Natural	Natural-cortes	R-and-F
bajo	gap vs blb (%)	7	12	8
	tiempo (s)	602,02	601,09	103,10
	mejora RnF (%)	-25	25	no aplica
medio	gap vs blb (%)	14	29	13
	tiempo (s)	600,57	600,15	552,33
	mejora RnF (%)	0	55	no aplica
alto	gap vs blb (%)	11	19	9
	tiempo (s)	602,01	600,18	496,46
	mejora RnF (%)	24	55	no aplica

Relax-and-Fix, que como fue visto, permite conocer que porcentaje de la brecha obtenida con Natural fue reducido al utilizar la heurística. La heurística logró cerrar la brecha de optimalidad hasta un 68 %, para el caso de la formulación Natural con cortes de solver, costo de configuración 125, y retornos medios que se presenta en la Tabla 6.15. Además, el tiempo de ejecución de la heurística fue ligeramente inferior en siete de los casos estudiados. Para los casos con retornos bajos, la heurística tuvo un menor desempeño. Como fue mencionado, el solver puede estar mejor preparado para resolver estos casos, que guardan mayor similitud con otros problemas estudiados de planificación de la producción sin retornos. Para costos de configuración 125, la formulación Natural tuvo un desempeño superior con y sin cortes de solver. Para costos de configuración 250, la heurística logró cerrar la brecha de optimalidad de Natural con cortes de solver en un 12 % en promedio. Para costos de configuración 500, la heurística logró cerrar la brecha de optimalidad de Natural sin cortes de solver en un 25 %. La diferencia en los tiempos de ejecución a favor de la heurística fue mayor para el caso de retornos bajos con costos de configuración 250 y 500.

Capítulo 7

Conclusiones

En este trabajo, fueron estudiadas dos variantes estocásticas de Problemas de Dimensionado de Lotes con Remanufactura, una con líneas de producción separadas para la manufacturación y remanufacturación (SELSRs), y otra con una única línea de producción compartida (SELSRj). En ambas variantes se incorpora incertidumbre en los valores de demanda y retornos.

Un estudio bibliográfico, que parte de los primeros modelos de dimensionado de lotes con y sin remanufactura hasta llegar a versiones que incorporan incertidumbre en los parámetros, permitió conocer diversas estrategias de modelado y resolución para distintas variantes. Dicho estudio sirvió de base para la formulación mediante programación entera-mixta de SELSRs y SELSRj utilizando árboles de escenarios.

Diversas estrategias de resolución alternativas fueron analizadas para la resolución computacional del problema SELSRs. A su vez, las mismas estrategias fueron desarrolladas a nivel teórico para SELSRj, y su implementación guarda una gran similitud con la del SELSRs. Las estrategias estudiadas fueron: diseño de una formulación extendida (reformulación localización de instalación no capacitada); reformulación incorporando inecuaciones válidas a nivel de camino; reformulación incorporando inecuaciones válidas de árbol y metodología de resolución heurística relax-and-fix.

7.1. Reformulación con inecuaciones válidas de camino

Inecuaciones (l, S, WW) presentadas para problemas deterministas primero por Retel Helmrich et al. (2014) y Cunha y Melo (2016) y luego ampliadas por Ali (2017) fueron adaptadas en este trabajo para los nuevos problemas SELSRs y SELSRj. El agregado de las inecuaciones

ciones adaptadas permite desarrollar la reformulación que incorpora inecuaciones válidas a nivel de camino. Se realizó un estudio computacional preliminar para analizar posibles combinaciones de inecuaciones válidas a agregar. Se observó que la incorporación de las nuevas inecuaciones adaptadas de Ali (2017) permiten mejorar las cotas inferiores obtenidas mediante la relajación a programación lineal del problema, con respecto a únicamente incorporar las adaptadas de Retel Helmrich et al. (2014) para instancias del problema SELSRs. Por este motivo, se selecciona la formulación que incorpora ambos tipos de inecuaciones para realizar el estudio computacional. Por otro lado, previo a la realización de las pruebas que se presentan en este trabajo, se comparó el desempeño de las formulaciones con inecuaciones equivalentes que tienen o no variables de inventario para instancias de tamaño pequeño y medio. Se observó un desempeño ligeramente superior para la formulación con variables de inventario.

7.2. Reformulación con inecuaciones válidas de árbol

Se estudiaron reformulaciones con inecuaciones válidas de árbol, que se desarrollan a través de la combinación de un conjunto de inecuaciones válidas de camino, con el objetivo de formar una nueva inecuación válida, posiblemente más ajustada. De modo de limitar los pares de inecuaciones que se combinan en nuestro problema, se utiliza una técnica similar a la estrategia de aproximación parcial propuesta por Vyve (2006). Se estudiaron formulaciones con inecuaciones de árbol combinando o bien familias de inecuaciones de camino con variables de inventario, o bien familias equivalentes sin variables de inventario. Analizando diversas combinaciones con la técnica mencionada anteriormente, se observó que el tiempo de armado del modelo por parte del solver al combinar inecuaciones con variables de inventario es muy superior al que se obtiene combinando las inecuaciones equivalentes sin variables de inventario.

Nos interesó determinar si el agregado de inecuaciones de árbol mejora las cotas inferiores del problema relajado a programación lineal, con respecto a la formulación que únicamente incorpora restricciones de camino. Utilizando la técnica mencionada, se observó que aún utilizando un único rango sobre distintas instancias de igual tamaño, se puede obtener mejoras en las relajaciones de la mayoría de las mismas. Se concluye que la combinación de restricciones de camino tiene el potencial de mejorar las formulaciones con inecuaciones válidas para el SELSRs. Por otro lado, la técnica mencionada no parece ser viable como estrategia de selección de inecuaciones, ya que requiere la incorporación de una cantidad importante de inecuaciones que impacta negativamente el desempeño del solver. Este motivo

imposibilitó comparar la formulación con restricciones de árbol con las otras formulaciones en el estudio computacional. Puede ser interesante estudiar mecanismos más sofisticados para incorporar únicamente restricciones de árbol que no sean redundantes, y así poder ser utilizadas en la práctica.

7.3. Heurística

Como complemento al estudio de las estrategias de resolución exacta presentadas anteriormente, se analizó una metodología de resolución heurística computacional para el SELSRs, la heurística relax-and-fix. Diversas estrategias de diseño fueron estudiadas en este trabajo. Estas estrategias se pueden agrupar en base a la forma en que se seleccionan los subconjuntos (conocidos como ventanas) en cada iteración para fijar, mantener integralidad, y relajar integralidad de las variables. Las principales estrategias consideradas fueron las siguientes. En primer lugar, se presenta una implementación que divide los subconjuntos de variables acorde al período al que pertenecen, y se tiene una ventana central que abarca una cantidad de períodos constante entre las iteraciones. Es decir, la cantidad de períodos al que pertenecen el subconjunto de variables que mantienen la restricción de dominio binario es constante. Esta implementación fue presentada en la Sección 5.2, y tiene el inconveniente de que cuando la ventana central abarca varios períodos, el método puede estancarse si se busca obtener soluciones óptimas en iteraciones intermedias. Dado que en este trabajo la heurística se utiliza sobre una variante con parámetros estocásticos, se analiza una nueva estrategia que tome en consideración la estructura de árbol del problema. En la segunda estrategia que se estudia en este trabajo en la Sección 5.3, se analiza la descomposición en subproblemas dentro de un período de tiempo. Esta estrategia es utilizada anteriormente en Alonso et al. (2000) y Beraldi et al. (2006) para distintos problemas con incertidumbre. Haciendo uso de la estructura de los problemas estudiados, los autores descomponen el problema a resolver dentro de una iteración en problemas independientes, lo que disminuye drásticamente la cantidad de variables enteras. En este trabajo, se muestra que la creación de subproblemas aplicando dicha estrategia sobre el problema SELSRs, no permite obtener problemas independientes. Asimismo, se sugiere un nuevo enfoque para obtener problemas independientes, que consiste en fijar en cada iteración no sólo las variables binarias, sino que también las variables continuas de nuestro problema. Finalmente, en la Sección 5.4, se analiza una tercer estrategia, para la que se diseña un algoritmo que permite tener una ventana central que abarque una cantidad de períodos que pueda variar entre las distintas iteraciones. Dado que la dificultad de resolución de los subproblemas depende en alta medida de la cantidad de variables binarias, este enfoque permite tener tamaños de ventanas que mantengan una cantidad de variables

binarias que se puedan resolver con las capacidades de cómputo que se disponen. Una experimentación preliminar con la implementación de la Sección 5.4, permite concluir que la misma se adapta de mejor manera al problema estudiado que las implementaciones anteriores, por lo que fue seleccionada para realizar el estudio computacional.

7.4. Dificultad de resolución de las instancias

Además de comparar el desempeño de las estrategias de resolución, puede ser interesante identificar las características del problema con mayor impacto en la resolución de las instancias. Para un tamaño de problema dado, la cantidad de variables binarias presentes en la instancia es un factor importante en la dificultad de resolución. De todos modos, se debe tener en cuenta que al comparar instancias con distintos tamaños de problema y una misma cantidad de variables binarias, el tiempo de ejecución en el procesamiento de un nodo del branch-and-bound puede tener una diferencia no despreciable debido a la cantidad de variables continuas. Esto se observa principalmente cuando la diferencia de tamaño de las instancias es grande. Para algunas implementaciones de la heurística, por ejemplo, se observa que incluso en los casos en que la cantidad de variables binarias en una etapa coincide para 8 y 12 períodos, el tiempo de resolución de la etapa generalmente es mayor para la instancia con mayor cantidad de períodos.

El tamaño del problema toma una mayor importancia cuando se busca resolver el problema mediante la formulación Natural y las reformulaciones. En estos casos, la cantidad de variables binarias aumenta exponencialmente al aumentar la cantidad de períodos en el horizonte de planificación. Por este motivo, mientras para casos de instancias de tamaño pequeño es posible obtener soluciones óptimas, para instancias de tamaño medio y grande esto no fue posible. De modo similar, mientras para las instancias de tamaño medio se obtienen soluciones factibles con facilidad, existen casos para los que alguna de las reformulaciones no pudo obtener soluciones factibles en el tiempo de ejecución asignado, debido al alto costo de generación del modelo por parte del solver.

Otro aspecto que parece tener un impacto claro en la dificultad de resolución es la cantidad de retornos. Se observa que al fijar el tamaño del problema y los costos de configuración, las instancias son en general más fáciles de resolver cuando la cantidad de retornos es baja. Como fue mencionado en el capítulo anterior, una posible explicación es que las instancias con pocos retornos pueden guardar una mayor similitud con el problema de dimensionado de lotes sin opciones de remanufactura. Asimismo, la mayor dificultad de resolver casos

con cantidad de retornos medios y altos también se puede explicar observando que en estos casos crece la cantidad de alternativas de solución, ya que existe una mayor posibilidad de alternar entre producción y remanufacturación. Al tener una mayor cantidad de alternativas de solución para considerar, el problema es más difícil de resolver.

7.5. Comparación de desempeño de estrategias de resolución

Para la mayoría de los casos de tamaño pequeño y medio, el solver pudo encontrar buenas soluciones en tiempos cortos de ejecución, para todas las formulaciones consideradas. Para estos casos, toma especial valor poder demostrar la cercanía al óptimo de la solución obtenida. Cuando es posible obtener una solución óptima con alguna reformulación, se observa que las reformulaciones logran probar la optimalidad de una solución más fácilmente que la formulación Natural. Las reformulaciones tienen relajaciones más próximas al casco convexo, lo que permitiría, para esos casos, cerrar más rápidamente la brecha de optimalidad con las mejores cotas inferiores que se obtienen.

Independientemente del tamaño del problema, las brechas de relajación a programación lineal de las dos reformulaciones son similares, siendo ligeramente inferiores para FL (formulación de localización de instalación). Las brechas de optimalidad, por otro lado, fueron mejores para IV (formulación con inecuaciones válidas) en la mayoría de los casos, cuando no se utilizan cortes de solver. Cuando estos se incorporan, la diferencia se vuelve aún más notoria. En muchos de estos casos, tanto las brechas de optimalidad, los tiempos de ejecución como el porcentaje de soluciones óptimas fueron mejores con la formulación que incorpora inecuaciones válidas de tipo (I, S, WW) . Una posible explicación es que dentro de los cortes del solver incorporados, se encuentran cortes de flujo. Dichos cortes pueden ser redundantes dado que la formulación FL es de por sí una representación de una red de flujo. De esta manera, el desempeño de la reformulación con los cortes incorporados puede estarse deteriorando.

Las reformulaciones que incorporan inecuaciones válidas de tipo (I, S, WW) y las de localización de instalación, han sido anteriormente estudiadas para versiones deterministas de problemas con remanufactura. Para el problema determinista, se observa que a medida que aumenta el tamaño de las instancias, en general mejora el desempeño de las reformulaciones al comparar con la formulación Natural. Para el problema estudiado, que incorpora incertidumbre en los parámetros de demanda y retornos, las reformulaciones permiten obtener buenas relajaciones del problema. Esta característica permite obtener soluciones óptimas en

tiempos cortos para tamaños de problema pequeños. A diferencia de lo que sucede para el problema determinista, aumentar el tamaño del problema puede significar un deterioro en el desempeño de las reformulaciones en comparación con Natural. En estos casos la gran cantidad de restricciones que presenta el modelo parece afectar el desempeño del solver, de modo que incluso en problemas con estructura de árboles completos de aridad 2 se observan diferencias en los resultados obtenidos con respecto a los de la literatura. A partir de determinados tamaños de problema, entonces, parece conveniente considerar la formulación Natural. A pesar de esto, incluso para tamaños de problemas grandes, las reformulaciones son valiosas ya que permiten obtener buenas cotas inferiores, de mejor calidad que la formulación Natural. Estos valores fueron de utilidad para comparar el desempeño de la heurística. En la práctica, las reformulaciones pueden ser de utilidad para conocer la cercanía de una solución obtenida mediante alguna estrategia de resolución. Puede ser interesante estudiar una manera más eficiente de incorporar a la formulación con inecuaciones válidas únicamente aquellas restricciones que no son dominadas por otras.

Para todos los tamaños de instancias, en la mayoría de los casos las reformulaciones presentan menores brechas de optimalidad y tiempos de ejecución en comparación con la formulación Natural cuando no se utilizaron cortes de solver. Sin embargo, para instancias de tamaño grande, se observa que los valores de función objetivo son ligeramente mejores para la formulación Natural. Mientras las reformulaciones permiten obtener buenas cotas inferiores, la obtención de estas cotas pueden perjudicar la obtención de mejores soluciones factibles.

El desempeño de la incorporación de los cortes de solver también se ve afectado al aumentar el tamaño del problema. Para instancias de tamaño pequeño, los cortes parecen permitir encontrar una solución óptima más rápidamente. A partir de tamaño medio, se observa que los cortes enlentecen la búsqueda de buenas soluciones factibles en algunas instancias. Por otro lado, cuando las instancias no son resueltas a optimalidad, tanto los cortes como el agregado de inecuaciones válidas tiene poco efecto sobre la mejor solución factible encontrada. La mayor ventaja en la utilización de los cortes del solver se observó para instancias con retornos bajos.

Al considerar cortes del solver, en general se observa una mejoría en las brechas de optimalidad de IV y Natural. Esta mejoría no es notoria para FL para tamaños de instancia medio y grande, existiendo casos en los que la inclusión de estos cortes devuelven peores brechas y valores de función objetivo. A medida que aumenta el tamaño de la instancia, se evidencia un deterioro en el desempeño de las reformulaciones IV y FL con la incorporación de cortes del

solver en el valor de función objetivo. Para FL, también se observa un deterioro en la brecha de optimalidad. Para instancias de tamaño medio, los mejores valores de función objetivo se obtienen con las reformulaciones sin utilizar cortes de solver en seis de los nueve casos estudiados. En los tres restantes, los mejores valores se obtienen con la formulación Natural, con cortes de solver.

En cuanto al desempeño de la heurística para tamaño medio, se obtuvieron valores de función objetivo cercanos a los de las formulaciones en tiempos de ejecución considerablemente menores, en especial para casos con retornos medio y alto. Para tamaño grande, el estudio se basó en analizar el porcentaje en que la heurística logra cerrar las brechas de optimalidad, calculadas a través de las mejores soluciones obtenidas con Natural, y las mejores cotas inferiores obtenidas con cualquiera de las formulaciones, acorde al cálculo de *Mejora Relax-and-Fix* de la Sección 6.5.2. Este cálculo nos permite estudiar el desempeño de la heurística, aún cuando las soluciones obtenidas por el solver son cercanas a la óptima. La heurística tuvo un desempeño superior a Natural en menores tiempos de ejecución, con y sin cortes de solver, para todos los casos con retornos medios y altos. La heurística logró cerrar la brecha de optimalidad hasta un 68%, para el caso de la formulación Natural con cortes de solver, costo de configuración 125, y retornos medios.

Capítulo 8

Trabajos Futuros

En este capítulo se presentan posibles extensiones para la presente tesis. En primer lugar, se presentan ideas para la generación de casos de estudio que complementen los ya considerados en este trabajo. En segundo lugar, se mencionan algunas modificaciones que es posible analizar para rediseñar las estrategias de resolución presentadas. Finalmente, se mencionan modificaciones sencillas a los modelos iniciales considerados, que permiten estudiar otro tipo de realidades.

Datos del problema y configuración de ejecución

Retel Helmrich et al. (2014) destacan que la resolución del problema determinista con líneas de producción separadas es más difícil de resolver en la práctica que el problema con una única línea de producción. Nuestro estudio computacional se concentró en el problema SELSRs, el cual posee el doble de variables binarias por sobre la versión con una única línea de producción SELSRj. Sin embargo, puede resultar interesante realizar un estudio análogo para analizar el desempeño de la formulación Natural, las reformulaciones y la heurística para dicho problema. Si bien se espera que su resolución sea más sencilla, la menor cantidad de variables puede permitir resolver más rápidamente problemas de tamaño grande. A su vez, se pueden diseñar casos de prueba con un horizonte de planificación mayor, sin necesidad de aumentar la capacidad de cómputo para la resolución.

Una de las principales limitantes de este estudio, fue la de no poder realizar pruebas de mayor tamaño, debido tanto al costo de la creación de los datos de las instancias, como al esfuerzo computacional necesario para la resolución. Contar con instancias con un mayor horizonte de planificación puede ser de utilidad para analizar en mayor detalle el desempeño de la heurística, ya que para esos casos se espera que las soluciones obtenidas con las formulaciones

no sean cercanas a la óptima. Aumentar la capacidad de cómputo es una estrategia posible para estudiar instancias con una mayor cantidad de períodos en el horizonte de planificación del problema. Otra posibilidad, es la de crear instancias cuya estructura de datos no sea la de un árbol completo. Esto se puede lograr fácilmente, por ejemplo, asumiendo que los eventos aleatorios suceden cada una determinada cantidad de períodos, a diferencia de en todos los períodos como se realiza en este trabajo. De esta manera, el tamaño del problema disminuye considerablemente, permitiendo aumentar el horizonte de planificación.

Considerar instancias de distinta aridad, es otra variación a parámetros del problema que puede ser estudiada. Asimismo, puede ser interesante diseñar casos de prueba con distintos niveles de correlación entre los parámetros aleatorios.

Estrategias de resolución

La utilización de inecuaciones (l, S, WW) en la formulación que agrega inecuaciones válidas permite trabajar con un subconjunto más manejable computacionalmente de inecuaciones (l, S) . Otra opción es la de desarrollar un algoritmo de separación para seleccionar más eficientemente las inecuaciones a agregar.

El agregado de inecuaciones de árbol permitió obtener mejoras en las cotas inferiores del problema relajado a programación lineal cuando se agrega una cantidad elevada de las mismas, con respecto a la formulación que únicamente incorpora restricciones de camino. Sin embargo, esta elevada cantidad de inecuaciones repercute negativamente en los tiempos de ejecución. Por lo tanto, es importante realizar un estudio de dominancia. De esta manera, se incorporan únicamente aquellas restricciones de árbol que no sean dominadas por otras inecuaciones, lo cual disminuye ampliamente su cantidad. Además de un estudio de dominancia, también se puede explorar algunas estrategias heurísticas para el agregado. Por ejemplo, en las instancias corridas se observó que generalmente las inecuaciones que lograban acercarse al casco convexo del problema, eran inecuaciones que se generaban seleccionando conjuntos de nodos que estaban cercanos en el árbol. Esta apreciación, u otras pueden ser utilizadas para generar métodos que permitan agregar inecuaciones con una mayor probabilidad de no ser redundantes.

Para la implementación con heurística relax-and-fix, es posible estudiar más a fondo distintas configuraciones de ejecución por sobre las estudiadas en el trabajo. Por ejemplo, si se utilizan soluciones subóptimas en las iteraciones intermedias, es importante analizar distintos tiempos de ejecución máximos para las mismas. También, se debe analizar que inecuaciones válidas

se eliminan en las últimas iteraciones, de modo de disminuír apropiadamente la cantidad de restricciones, y obtener soluciones más rápidamente. Finalmente, también es conveniente analizar la conveniencia de fijar variables contínuas, y en caso afirmativo, determinar cuales de estas seleccionar.

Extensiones del problema

Una posible extensión al problema estudiado, es la de incorporar el desmontaje de los retornos. En la literatura existen estudios de problemas con incertidumbre en los que se modela el desmontaje, aunque de nuestro conocimiento se trata de problemas en los que no se admite la posibilidad de satisfacer la demanda tanto por productos nuevos, como por productos remanufacturados.

Otra posible extensión, es la de considerar la disposición de retornos. En esta extensión, los retornos pueden ser desechados por motivos de exceso o calidad. Esta posibilidad puede permitir obtener soluciones de valor menor a las obtenidas, ya que se puede ahorrar los costos de almacenamiento y configuración de la producción para el SELSRj.

Para culminar, también es posible, además de considerar la cantidad de retornos recibidos, modelar la calidad de los mismos. De esta manera, se puede extender el problema para considerar retornos heterogéneos.

Bibliografía

- Aggarwal, A. y Park, J. K. Improved algorithms for economic lot size problems. *Operations Research*, 41(3):549–571, 1993.
- Ahmed, S., King, A. J., y Parija, G. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Journal of Global Optimization*, 26(1):3–24, 2003.
- Ali, S. A. S. A computational analysis of lower bounds for the economic lot sizing problem in remanufacturing with separate setups. *Journal of Physics: Conference Series*, 890(1): 012098, 2017.
- Alonso, A., Escudero, L. F., y Ortuno, M. T. A stochastic 0–1 program based approach for the air traffic flow management problem. *European Journal of Operational Research*, 120(1):47–62, 2000.
- Baki, M. F., Chaouch, B. A., y Abdul-Kader, W. A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery. *Computers & Operations Research*, 43:225 – 236, 2014.
- Barany, I., Van Roy, T., y Wolsey, L. A. *Uncapacitated lot-sizing: The convex hull of solutions*, pp. 32–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984a.
- Barany, I., Van Roy, T., y Wolsey, L. A. Strong formulations for multi-item capacitated lot sizing. *Management Science*, 30(10):1255–1261, 1984b.
- Beale, E. M. L. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(2):173–184, 1955.
- Beale, E., Forrest, J., y Taylor, C. Multi-time-period stochastic programming. *Stochastic programming*, pp. 387–402, 1980.
- Beltrán, J. L. y Krass, D. Dynamic lot sizing with returning items and disposals. *IIE Transactions*, 34(5):437–448, 2002.
- Benders, J. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
- Beraldi, P., Ghiani, G., Grieco, A., y Guerriero, E. Fix and relax heuristic for a stochastic lot-sizing problem. *Computational Optimization and Applications*, 33(2-3):303–318, 2006.

- Bilde, O. y Krarup, J. Sharp lower bounds and efficient algorithms for the simple plant location problem. In Hammer, P., Johnson, E., Korte, B., y Nemhauser, G., editors, *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pp. 79–97. Elsevier, 1977.
- Birge, J. R. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985.
- Birge, J. R. y Louveaux, F. *Introduction to Stochastic Programming*. Springer Publishing Company, Incorporated, 2nd edition, 2011.
- Brahimi, N., Absi, N., Dauzere-Peres, S., y Nordli, A. Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3):838 – 863, 2017.
- Chen, M. y Abrishami, P. A mathematical model for production planning in hybrid manufacturing-remanufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 71(5-8):1187–1196, 2014.
- Cunha, J. O. y Melo, R. A. A computational comparison of formulations for the economic lot-sizing with remanufacturing. *Computers & Industrial Engineering*, 92:72 – 81, 2016.
- Dantzig, G. B. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.
- Dantzig, G. B. y Wolfe, P. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.
- Devoto, C., Fernández, E., y Piñeyro, P. The economic lot-sizing problem with remanufacturing and inspection for grading heterogeneous returns. *Journal of Remanufacturing*, 11:71 - 87, 2021.
- Dixon, P. S. y Silver, E. A. A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. *Journal of Operations Management*, 2(1):23 – 39, 1981.
- Eppen, G. D. y Martin, R. K. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, 35(6):pp. 832–848, 1987.
- Fang, C., Liu, X., Pardalos, P., Long, J., Pei, J., y Zuo, C. A stochastic production planning problem in hybrid manufacturing and remanufacturing systems with resource capacity planning. *Journal of Global Optimization*, 68, 2017.
- Federgruen, A. y Tzur, M. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n \log n)$ or $O(n)$ time. *Management Science*, 37(8):909–925, 1991.
- Fleischmann, B. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, 44(3):337–348, 1990.
- Florian, M., Lenstra, J. K., y Kan, A. H. G. R. Deterministic production planning: Algorithms and complexity. *Management Science*, 26(7):669–679, 1980.

- Florian, M. y Klein, M. Deterministic production planning with concave costs and capacity constraints. *Management Science*, 18(1):12–20, 1971.
- Garey, M. R. y Johnson, D. S. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- Gassmann, H. I. Mslip: A computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.
- Golany, B., Yang, J., y Yu, G. Economic lot-sizing with remanufacturing options. *IIE Transactions*, 33:995–1004, 2001.
- Gopalakrishnan, M., Ding, K., Bourjolly, J.-M., y Mohan, S. A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, 47(6): 851–863, 2001.
- Guan, Y. y Miller, A. J. Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research*, 56:1172–1183, 2008.
- Guan, Y., Ahmed, S., Miller, A. J., y Nemhauser, G. L. On formulations of the stochastic uncapacitated lot-sizing problem. *Operations Research Letters*, pp. 241–250, 2006a.
- Guan, Y., Ahmed, S., y Nemhauser, G. L. *Cutting planes for multi-stage stochastic integer programs*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, 2006b.
- Guan, Y., Ahmed, S., Nemhauser, G. L., y Miller, A. J. A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. *Mathematical Programming*, 105(1):55–84, 2006c.
- Guan, Y., Ahmed, S., y Nemhauser, G. L. Sequential pairing of mixed integer inequalities. *Discrete Optimization*, 4(1):21 – 39, 2007. Mixed Integer Programming.
- Harris, F. How many parts to make at once. *Operations Research*, 38:947–950, 1913.
- Haugen, K. K., LÄÅ_kketangen, A., y Woodruff, D. L. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research*, 132 (1):116 – 122, 2001.
- Hilger, T., Sahling, F., y Tempelmeier, H. Capacitated dynamic production and remanufacturing planning under demand and return uncertainty. *OR Spectrum*, 38(4):849–876, 2016.
- Homrich, A. S., Galvao, G., Abadia, L. G., y Carvalho, M. M. The circular economy umbrella: Trends and gaps on integrating pathways. *Journal of Cleaner Production*, 175:525–543, 2018.
- Hsu, V. N. Dynamic Economic Lot Size Model with Perishable Inventory. *Management Science*, 46(8):1159–1169, 2000.
- Hsu, W.-L. On the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine. *Management Science*, 29(1):93–105, 1983.

- Ijomah, W. *A Model-based Definition of the Generic Remanufacturing Business Process*. PhD thesis, 2002.
- Junior, M. L. y Filho, M. G. Master disassembly scheduling in a remanufacturing system with stochastic routings. *Central European Journal of Operations Research*, 25(1):123–138, 2017.
- Kilic, O. A., Tunc, H., y Tarim, S. A. Heuristic policies for the stochastic economic lot sizing problem with remanufacturing under service level constraints. *European Journal of Operational Research*, 267(3):1102–1109, 2018.
- Koken, P., Arasanipalai Raghavan, V., y Yoon, S. W. A genetic algorithm based heuristic for dynamic lot sizing problem with returns and hybrid products. *Computers & Industrial Engineering*, 119:453–464, 2018.
- Loparic, M., Pochet, Y., y Wolsey, L. A. The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming*, 89(3):487–504, 2001.
- Lozano, S., Larraneta, J., y Onieva, L. Primal-dual approach to the single level capacitated lot-sizing problem. *European Journal of Operational Research*, 51(3):354 – 366, 1991.
- Macedo, P. B., Alem, D., Santos, M., Junior, M. L., y Moreno, A. Hybrid manufacturing and remanufacturing lot-sizing problem with stochastic demand, return, and setup costs. *The International Journal of Advanced Manufacturing Technology*, 82(5):1241–1257, 2016.
- Maes, J. y Wassenhove, L. V. Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review. *The Journal of the Operational Research Society*, 39(11):pp. 991–1004, 1988.
- Maes, J., McClain, J. O., y Wassenhove, L. N. V. Multilevel capacitated lotsizing complexity and lp-based heuristics. *European Journal of Operational Research*, 53(2):131 – 148, 1991.
- Mahapatra, S., Pal, R., y Narasimhan, R. Hybrid (re)manufacturing: manufacturing and operational implications. *International Journal of Production Research*, 50(14):3786–3808, 2012.
- Manne, A. S. Programming of economic lot sizes. *Management Science*, 4(2):115–135, 1958.
- Naeem, M. A., Dias, D. J., Tibrewal, R., Chang, P. C., y Tiwari, M. K. Production planning optimization for manufacturing and remanufacturing system in stochastic environment. *Journal of Intelligent Manufacturing*, 24(4):717–728, 2013.
- Nahmias, S. Perishable inventory theory: A review. *Operations Research*, 30:680–708, 1982.
- Nemhauser, G. L. y Wolsey, L. A. *Integer and combinatorial optimization*. John Wiley & Sons, 1988.
- Pan, Z., Tang, J., y Liu, O. Capacitated dynamic lot sizing problems in closed-loop supply chain. *European Journal of Operational Research*, 198(3):810–821, 2009.

- Piñeyro, P. y Viera, O. Inventory policies for the economic lot-sizing problem with remanufacturing and final disposal options. *Journal of Industrial and Management Optimization*, 5:217–238, 2009.
- Piñeyro, P. y Viera, O. The economic lot-sizing problem with remanufacturing and one-way substitution. *International Journal of Production Economics*, 124(2):482 – 488, 2010.
- Pochet, Y. y Wolsey, L. A. Polyhedra for lot-sizing with wagner-whitin costs. *Mathematical Programming*, 67(1):297–323, 1994.
- Pochet, Y. y Wolsey, L. A. *Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- Quezada, F., Gicquel, C., Kedad-Sidhoum, S., y Vu, D. Q. A multi-stage stochastic integer programming approach for a multi-echelon lot-sizing problem with returns and lost sales. *Computers & Operations Research*, 116, 2020.
- Reike, D., Vermeulen, W. J., y Witjes, S. The circular economy: new or refurbished as ce 3.0? exploring controversies in the conceptualization of the circular economy through a focus on history and resource value retention options. *Resources, Conservation and Recycling*, 135:246–264, 2018.
- Retel Helmrich, M., Jans, R., van den Heuvel, W., y Wagelmans, A. Economic lot-sizing with remanufacturing: complexity and efficient formulations. *IIE Transactions*, 46(1): 67–86, 2014.
- Richter, K. y Sombrutzki, M. Remanufacturing planning for the reverse wagner-whitin models. *European Journal of Operational Research*, 121(2):304 – 315, 2000.
- Richter, K. y Weber, J. The reverse Wagner-Whitin model with variable manufacturing and remanufacturing cost. *International Journal of Production Economics*, 71(1-3):447–456, 2001.
- Rockafellar, R. T. y Wets, R. J.-B. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.
- Rogers, J. A computational approach to economic-lot scheduling problem. *Management Science*, 4:264–291, 1958.
- Römisch, W. y Schultz, R. *Multistage Stochastic Integer Programs: An Introduction*, pp. 581–600. 2001.
- Roshani, A., Giglio, D., y Paolucci, M. A simulated annealing approach for the capacitated dynamic lot sizing problem in a closed remanufacturing system. *International Federation of Automatic Control - PapersOnLine*, 49:1496–1501, 2016.
- Roshani, A., Giglio, D., y Paolucci, M. A relax-and-fix heuristic approach for the capacitated dynamic lot sizing problem in integrated manufacturing/remanufacturing systems. *IFAC-PapersOnLine*, 50(1):9008 – 9013, 2017. 20th IFAC World Congress.

- Sahling, F. A column-generation approach for a short-term production planning problem in closed-loop supply chains. *Business Research*, 6(1):55–75, 2013.
- Schulz, T. A new silver meal based heuristic for the single-item dynamic lot sizing problem with returns and remanufacturing. *International Journal of Production Research*, 49(9): 2519–2533, 2011.
- Sen, S. Algorithms for stochastic mixed-integer programming models. In Aardal, K., Nemhauser, G., y Weismantel, R., editors, *Discrete Optimization*, volume 12 of *Handbooks in Operations Research and Management Science*, pp. 515 – 558. Elsevier, 2005.
- Sifaleras, A. y Konstantaras, I. Variable neighborhood descent heuristic for solving reverse logistics multi-item dynamic lot-sizing problems. *Computers & Operations Research*, 78: 385–392, 2017.
- Sifaleras, A., Konstantaras, I., y Mladenovic, N. Variable neighborhood search for the economic lot sizing problem with product returns and recovery. *International Journal of Production Economics*, 160:133 – 143, 2015.
- Suzanne, E., Absi, N., y Borodin, V. Towards circular economy in production planning: Challenges and opportunities. *European Journal of Operational Research*, 287(1):168 – 190, 2020.
- Taft, E. The most economical production lot. *The Iron Age*, 101:1410–1412, 1918.
- Tang, L., Che, P., y Liu, J. A stochastic production planning problem with nonlinear cost. *Computers & Operations Research*, 39:1977–1987, 2012.
- Tempelmeier, H. y Derstroff, M. A lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. *Management Science*, 42(5):738–757, 1996.
- Teunter, R. H., Bayindir, Z. P., y Heuvel, W. V. D. Dynamic lot sizing with product returns and remanufacturing. *International Journal of Production Research*, 44(20):4377–4400, 2006.
- Tintner, G. Stochastic linear programming with applications to agricultural economics. In *Proceedings of the Second Symposium in Linear Programming*, volume 1, pp. 197–228. National Bureau of Standards Washington, DC, 1955.
- Trigeiro, W. W., Thomas, L. J., y McClain, J. O. Capacitated lot sizing with setup times. *Management Science*, 35(3):353–366, 1989.
- van de Panne, C. y Popp, W. Minimum-cost cattle feed under probabilistic protein constraints. *Management Science*, 9(3):405–430, 1963.
- Van den Heuvel, W. *The economic lot-sizing problem : new results and extensions*. PhD thesis, 2006.
- Van Slyke, R. M. y Wets, R. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

- Vargas, V. An optimal solution for the stochastic version of the wagner-whitin dynamic lot-size model. *European Journal of Operational Research*, 198(2):447 – 451, 2009.
- Vu, Q. D., Gicquel, C., y Kedad-Sidhoum, S. Stochastic Programming approaches for planning re-manufacturing activities under uncertain demand and returns forecasts . In *ROADEF 2017 - 18e conférence de la société française de Recherche Opérationnelle et d'Aide à la décision*, Metz, France, 2017.
- Vyve, L. A., Mathieu Vanand Wolsey. Approximate extended formulations. *Mathematical Programming*, 105(2):501–522, 2006.
- Wagelmans, A., Hoesel, S., y Kolen, A. Economic lot sizing: An $o(n \log n)$ algorithm that runs in linear time in the wagner-whitin case. *Maastricht University, Open Access publications from Maastricht University*, 40, 1992.
- Wagner, H. M. y Whitin, T. M. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.
- Yang, J., Golany, B., y Yu, G. A concave-cost production planning problem with remanufacturing options. *Naval Research Logistics*, 52(5):443–458, 2005.
- Yanikoğlu, I. y Denizel, M. The value of quality grading in remanufacturing under quality level uncertainty. *International Journal of Production Research*, 59(3):839–859, 2021.
- Zangwill, W. I. Minimum concave cost flows in certain networks. *Management Science*, 14(7):429–450, 1968.
- Zangwill, W. I. A backloging model and a multi-echelon model of a dynamic economic lot size production system—a network approach. *Management Science*, 15(9):506–527, 1969.
- Zhang, Z.-H., Jiang, H., y Pan, X. A lagrangian relaxation based approach for the capacitated lot sizing problem in closed-loop supply chain. *International Journal of Production Economics*, 140(1):249–255, 2012.
- Zhao, C. y Guan, Y. Extended formulations for stochastic lot-sizing problems. *Operations Research Letters*, 42(4):278 – 283, 2014.
- Zhou, Z. y Guan, Y. Stochastic lot-sizing problem with deterministic demands and wagner-whitin costs. *Operations Research Letters*, 38(5):414 – 419, 2010.
- Zouadi, T., Yalaoui, A., y Reghioui, M. Hybrid manufacturing/remanufacturing lot-sizing and supplier selection with returns, under carbon emission constraint. *International Journal of Production Research*, 56(3):1233–1248, 2018.