

UNIVERSIDAD DE CUENCA

FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL



“OPTIMIZACIÓN DE CORTE DE VARILLAS DE ACERO DE CONSTRUCCIÓN”

Monografía previa a la obtención del
título de Ingeniero Civil

AUTOR:

Gustavo Israel Ochoa Reiban

DIRECTOR:

PhD. Ing. Rodrigo Sempértegui

CUENCA – ECUADOR

2014



RESUMEN

El presente trabajo de investigación utiliza herramientas de modelación matemática para la optimización de corte de varillas de acero de construcción afianzada en la metodología de la Investigación de Operaciones. En este documento se da a conocer fundamentos teóricos aplicados en el presente proyecto a lo largo de diferentes autores que han escrito sobre la Investigación de Operaciones, características, sistemas, formulaciones etc. con la finalidad de conseguir un fundamento para la correcta elaboración de la investigación presente. Posteriormente, se construye un modelo matemático para la optimización de tipo Programación Lineal Entera Pura que tiene por fin, realizar cortes sobre barras de longitud estándar a partir de piezas demandadas de tal manera que se empleen el menor número de barras estándar, lo que repercute en una reducción del desperdicio y ahorro de dinero. Dicho modelo se sustenta en el principio de Patrones de Corte Unidimensional. Una vez finalizada la modelación se procede a la implementación informática del modelo mediante el software `MATLAB`. La herramienta informática diseñada denominada `LCO` permitirá ajustar al modelo de optimización de corte de varillas de construcción a diversos casos lo cual ayudara a determinar el tipo de varilla de longitud estándar que se tendrá que adquirir así como indicar los esquemas o patrones de corte óptimos para cortar las barras. Finalmente, en virtud de la necesidad de validar el modelo construido, se aplicará el programa `LCO` a un caso real.

Palabras clave: Investigación de Operaciones, optimización, modelación matemática, LCO, corte de varillas, MATLAB.



ABSTRACT

The present research work uses mathematical modeling tools to improve the cutting building steel rods supported by Methodology of Operations Research. Also, this paper exposes the theoretical foundations described by several authors about operation research, functions, systems, formulations, etc., which have been applied during the development of this research to get the enough academic support to prepare this work correctly. Subsequently, a mathematical model to optimized Pure Integer Linear Programming type will be built in order to make cuts on standard size bars through defended parts, such that the smallest number of standard bars will be used, which impacts on waste reduction and helps save money. Furthermore, this model is based on the principle of One-Dimensional Cutting Patterns. After the modeling has been completed, the implementation of the model will be performed through the **MATLAB** software. The computer tool called **LCO** will allow to the optimization model of cutting steel rods to fit in many cases, helping not only to define what kind of standard size rod will be needed to purchase but also to indicate the optimal schemes or patterns for cutting the bars. Finally, due to the need to validate the model developed, the **LCO** program will be applied to a real case

Keywords: Operations Research, optimization, mathematical modeling, **LCO**, cutting rods, **MATLAB**.



ÍNDICE GENERAL

RESUMEN	1
ABSTRACT	2
ÍNDICE DE ILUSTRACIONES	6
ÍNDICE DE TABLAS	7
CAPITULO I	12
INTRODUCCIÓN	12
1.2 ALCANCE	14
1.3 OBJETIVOS	14
1.3.1 General	14
1.3.2 Específicos	14
1.4 ESTRUCTURA DE LA MEMORIA	14
CAPITULO II	15
MARCO TEÓRICO.....	15
2.1 INVESTIGACIÓN OPERATIVA.....	15
2.2 MODELOS MATEMÁTICOS	16
2.3 CONSTRUCCIÓN DE UN MODELO	17
2.3.1 Identificación, Descripción y Características del Problema	17
2.3.2 Especificación y Formulación Matemática.....	17
2.3.3 Resolución.....	17
2.3.4 Verificación, validación y refinamiento.....	18
2.3.5 Interpretación y análisis de resultados.....	18
2.3.6 Mejoras.....	18
2.3.7 Implantación, documentación y mantenimiento.....	18
2.4 CUTTING AND PACKING PROBLEMS (C&P).....	18
2.4.1 Problema de la Mochila (Knapsack Problem).....	19



- 2.4.2 Problema de la Caja de Embalaje (Bin Packing Problem)..... 20
- 2.4.3 Problema de Corte de Stock (Cutting Stock Problem)..... 20
- 2.5 CORTE DE STOCK (CSP)..... 20
 - 2.5.1 Problema de Corte Unidimensional (1D CSP) 20
 - 2.5.1.1 Aplicaciones 1D CSP..... 21
- 2.6 PROGRAMACIÓN LINEAL 24
 - 2.6.1 Programación Lineal Entera 25
 - 2.6.1.1 Clasificación de la Programación Entera 26
- CAPITULO III 27
- CONSTRUCCIÓN DEL MODELO MATEMÁTICO 27
 - 3.1 IDENTIFICACIÓN, DESCRIPCIÓN Y CARACTERÍSTICAS DEL PROBLEMA 28
 - 3.1.1 Planteamiento del Modelo 28
 - 3.1.2 Objetivos del Modelo. 30
 - 3.1.3 Restricciones del Modelo..... 30
 - 3.1.4 Función Objetivo..... 30
 - 3.1.5 Variables decisión..... 30
 - 3.2 ESPECIFICACIÓN Y FORMULACIÓN MATEMÁTICA..... 31
 - 3.2.1 Restricciones de la Matriz de Patrones..... 32
 - 3.2.2 Patrones de Corte..... 33
 - 3.2.2.1 Obtención de Patrones de Corte 33
 - 3.2.3 Matriz de Posibles Patrones de Corte Eficiente *mppce* 35
 - 3.3 RESOLUCIÓN DEL MODELO..... 35
 - 3.3.1 Método de Anclado y Ramificado para la Conformación de Patrones de Corte..... 35
 - 3.3.2 Matriz de Patrones de Corte Eficiente 36
 - 3.3.3 Optimización Mediante Matlab..... 37
 - 3.3.4 Función Linprog..... 38
 - 3.3.4.1 Notación Linprog 38
 - 3.3.5 Optimización Linprog y Branch and Bound..... 39
 - 3.3.6 Solución Linprog y Branch and Bound..... 39
- CAPITULO IV 42



IMPLEMENTACION Y RESULTADOS DE LA HERRAMIENTA INFORMATICA

LCO 42

4.1 ESTRUCTURA DEL PROGRAMA 43

4.2 USO DE LCO 44

4.2.1 Datos de entrada 44

4.2.1.1 Generales 44

4.2.1.2 Piezas Demandadas 45

4.2.1 Ejecutar LCO 46

4.2.2 Solución LCO 47

4.3 RESULTADOS 49

4.3.1 Piezas de Acero Demandadas 49

4.3.2 Optimización mediante Programa LCO 51

4.3.2.1 Resultados mediante LCO 51

CAPITULO V 56

CONCLUSIONES 56

5.1 Referente a la Modelación 56

5.2 Referente al Programa LCO 57

5.3 Referente a la Investigación de Operaciones 58

5.3 Referente a los Objetivos Planteados 59

BIBLIOGRAFIA 60

ANEXOS 63

1 Código fuente de LCO en lenguaje Matlab 63

1.1 Función principal para optimización de corte de varillas de acero de construcción 63

1.2 Obtención de la matriz de Posibles Patrones de corte eficiente *mppce* 63

1.3 Solución del Modelo Matemático 68

1.4 Linprog y Branch and Bound (El presente código de la función “Branchandbound” pertenece a Sherif A. Tawfik, Faculty of Engineering, Cairo University para Problemas Lineales Enteros Puros) 68

1.5 Solución final: Patrones de corte eficientes, Porcentaje de Desperdicio, Barras a emplear 69



ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1: Tipos de Corte. Fuente: Autor.....</i>	19
<i>Ilustración 2: Esquemas de Patrones de Corte. Fuente: Autor.....</i>	29
<i>Ilustración 3 Posibles Patrones de Corte. Fuente: Autor.....</i>	34
<i>Ilustración 4: Posibles Patrones de Coste Eficiente. Fuente: Autor.....</i>	34
<i>Ilustración 5 Diagrama del Algoritmo Solución de la Matriz de Posibles Patrones de Corte Eficiente. Fuente: Autor</i>	36
<i>Ilustración 6: Diagrama del Algoritmo Solución del Modelo Matemático por Linprog B &B. Fuente: Autor</i>	41
<i>Ilustración 7: Diagrama de la herramienta informática LCO. Fuente: Autor</i>	43
<i>Ilustración 8: Datos generales a introducir en LCO</i>	44
<i>Ilustración 9: Datos de Piezas Demandadas a introducir en LCO.....</i>	45
<i>Ilustración 10: Tipos de Doblados de Hierros.....</i>	46
<i>Ilustración 11: Soluciones que puede ejecutar LCO.....</i>	46
<i>Ilustración 12: Salida de resultados de la optimización en LCO.....</i>	47
<i>Ilustración 13: Solución para satisfacer las necesidades logrando el menor desperdicio posible en LCO</i>	47
<i>Ilustración 14: Interfaz de la herramienta informática LCO</i>	48



ÍNDICE DE TABLAS

<i>Tabla 1: Lista de piezas de acero solicitadas. Fuente: Autor</i>	<i>50</i>
<i>Tabla 2: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 8mm mediante LCO.....</i>	<i>51</i>
<i>Tabla 3: Esquemas de corte óptimo para piezas de 8mm a partir de una longitud comercial de 12m mediante LCO</i>	<i>51</i>
<i>Tabla 4: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 10 mm mediante LCO.....</i>	<i>52</i>
<i>Tabla 5: Esquemas de corte óptimo para piezas de 10mm a partir de una longitud comercial de 12m mediante LCO</i>	<i>52</i>
<i>Tabla 6: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 14 mm mediante LCO.....</i>	<i>53</i>
<i>Tabla 7: Esquemas de corte óptimo para piezas de 14mm a partir de una longitud comercial de 12m mediante LCO</i>	<i>53</i>
<i>Tabla 8: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 16 mm mediante LCO.....</i>	<i>54</i>
<i>Tabla 9: Esquemas de corte óptimo para piezas de 16mm a partir de una longitud comercial de 6m mediante LCO</i>	<i>54</i>
<i>Tabla 10: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 20 mm mediante LCO.....</i>	<i>55</i>
<i>Tabla 11: Esquemas de corte óptimo para piezas de 20mm a partir de una longitud comercial de 12m mediante LCO</i>	<i>55</i>



Universidad de Cuenca
Clausula de propiedad intelectual

Yo, Gustavo Israel Ochoa Reiban, autor de la Monografía "Optimización de Corte de Varillas de Acero de Construcción", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 17 de junio de 2014

Gustavo Israel Ochoa Reiban

C.I: 0104001037



Universidad de Cuenca
Clausula de derechos de autor

Yo, *Gustavo Israel Ochoa Reiban*, autor de la Monografía "Optimización de Corte de Varillas de Acero de Construcción", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero Civil. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor

Cuenca, 17 de junio de 2014



Gustavo Israel Ochoa Reiban

C.I: 0104001037



A Cumanda, mi madre, por su coraje.



Agradezco a la vida, por rodearme de gente igual o más capaz que yo.

Entre ellos

Rigoberto.

Rosa Margarita.

Daniela.

Rodrigo.



CAPITULO I

INTRODUCCIÓN

El mundo de la construcción avanza a paso vertiginoso en términos tecnológicos, metodologías, recursos, alcance, etc. Es así, que prácticamente van de la mano del crecimiento económico de un país. Concretamente, en Ecuador en el año 2013 el PIB llegó al 4.5% del cual el 0.87% pertenece a la industria de la construcción tanto del sector público como privado, siendo este el contribuyente al PIB que más aporta de todas las industrias del país incluso superior a la industria de petróleo y minas con el 0.51% (Diario El Universo, 2014). Esto demuestra la influencia e importancia de este sector en el desarrollo de un país.

La industria de la construcción como actividad económica representa un motor de desarrollo, es una fuente generadora de empleo entre otros beneficios pero también demanda gran cantidad de recursos para la ejecución de las diversas obras. Gran parte de estos recursos no son renovables y teniendo en cuenta que prácticamente la mayoría de los materiales utilizados son de alto impacto para el medio ambiente. Para tener en cuenta, *en términos estadísticos, se puede decir que el sector de la construcción es responsable del 50% de los recursos naturales empleados, del 40%*



de la energía consumida y del 50% del total de los residuos generados (Arenas, 2008).

Como se puede observar el mundo de la construcción, está rodeado de problemas relacionados con el mal aprovechamiento de recursos, lo cual genera desperdicios y por ende pérdidas económicas y sobre todo impacto ambiental. Existen porcentajes de desperdicios aceptables que varían dependiendo el rubro en las construcciones, lo cual ha ayudado a pensar o creer a los constructores que existen desperdicios que están correctos debido a que se encuentran ya sea dentro o cerca en la mayoría de los casos de los rangos aceptables, produciendo una idea de que los desperdicios en la construcción son inmanejables. Específicamente, de acuerdo con (Soibelman, 2010) se generan aproximadamente entre el 7% y el 25% de desperdicios de acero de construcción. Específicamente en las varillas empleadas para el armado de las diferentes secciones que conforman una determinada estructura. Este tipo de desperdicios en gran parte se refiere a desperdicios provocados por cortes ineficientes de las varillas de construcción.

En vista de lo mencionado anteriormente, se hace imperioso desarrollar formas, sistemas, técnicas etc. que nos permita reducir lo más posible dichos desperdicios mencionados en obras civiles, teniendo en cuenta lógicamente que será prácticamente imposible evitar completamente los desperdicios. Por ello nos hemos enfocado en el presente proyecto específicamente a tratar los desperdicios producidos en el corte de varillas de acero de construcción.

Con la finalidad de mejorar el aspecto relacionado al desperdicio en los cortes de las varillas de acero de construcción, la presente investigación está apoyado en una rama nueva que nace a partir de la Segunda Guerra Mundial y debe su nombre a las operaciones militares como lo es la Investigación de Operaciones. Esta disciplina relativamente moderna que incorpora modelos matemáticos, estadísticos y algoritmos para modelar y resolver problemas complejos, determinando la solución óptima y mejorando la toma de decisiones sea el campo en el cual se esté implementando (PHPSimplex, 2006). Dentro de esta gran rama que tiene como función principal optimizar procesos, existe la rama de problemas denominada corte unidimensional en el cual se afianzara la investigación planteada. Todo esto con la finalidad de construir un modelo matemático eficiente, que nos permita optimizar el proceso de corte de varillas.

Para esto, se construirá un modelo matemático óptimo que a su vez será la base para diseñar un programa utilizando las bondades del software **MATLAB**. Esto permitirá automatizar el proceso, el cual ayudará a emplear el modelo en diferentes casos de estudio. Este programa tiene por fin, optimizar el proceso en estudio de tal manera que permita cortar las varillas a partir de los cortes sugeridos en las planillas de hierros de los planos estructurales, con ello se determina los patrones de corte eficientes que permitan reducir en lo posible los desperdicios y sugerir la mejor



varilla comercial a comprar para el proyecto, todo esto con el fin de reducir las pérdidas económicas e impacto ambiental.

1.2 ALCANCE.

La presente investigación tiene por finalidad construir una herramienta informática que permita optimizar el corte de varillas de acero de construcción para lo cual se diseñará un modelo matemático fundamentado en la investigación de operaciones, ésta aplicación será implementada a un caso real. Dicha herramienta será creada a través del software `MATLAB`.

1.3 OBJETIVOS

La presente monografía persigue los siguientes objetivos:

1.3.1 General.

Construir un programa que permita optimizar el corte de las varillas de acero de construcción.

1.3.2 Específicos.

1. Formular el modelo matemático a optimizar.
2. Diseñar un algoritmo de programación para el modelo matemático.
3. Validar el Modelo

1.4 ESTRUCTURA DE LA MEMORIA

La presente investigación es de tipo descriptiva, experimental, aplicada.

El presente trabajo se encuentra desarrollado a través de cinco capítulos incluido este. En los cuales se abordan las diferentes temáticas con la finalidad de llevar adelante los objetivos planteados. El Capítulo II se centra en la recopilación de fundamentos teóricos aplicados en el presente proyecto a lo largo de diferentes autores que han escrito sobre la investigación de operaciones, características, sistemas, formulaciones etc. con la finalidad de conseguir un fundamento para la correcta elaboración de la investigación presente. Finalizado esto, en el Capítulo III se lleva a cabo la construcción del modelo a optimizar en el cual se da a conocer las características del problema, la formulación y la resolución del mismo mediante el software `MATLAB`. Luego, en el Capítulo IV se da a conocer la herramienta Informática diseñada para optimizar el corte de varillas, donde se indicara cómo funciona el programa y obviamente con la finalidad de validar el modelo se realiza una aplicación al mundo real de la herramienta. Finalmente en el Capítulo V se dará a conocer las diversas conclusiones a las que se llegó finalizado el presente trabajo.



CAPITULO II

MARCO TEÓRICO

2.1 INVESTIGACIÓN OPERATIVA

La Investigación Operativa o de Operaciones (IO) también denominada ciencia de la administración (Winston, 2005) incorpora modelos matemáticos, estadísticos y algoritmos para modelar y resolver problemas complejos, determinando la solución óptima y mejorando la toma de decisiones sea el campo en el cual se esté implementando (PHPSimplex, 2006).

Esta disciplina relativamente moderna nace a partir de ciertos servicios militares que se prestaron a inicios de la Segunda Guerra Mundial. En aquel entonces existía el apremio de asignar recursos escasos a las distintas maniobras militares así como, las diversas acciones que conformaban cada operación de la manera más eficaz.



De esta manera tanto las administraciones militares británicas como estadounidenses con ayuda de científicos a quienes les ordenó hacer investigación sobre operaciones (militares) para problemas estratégicos y bélicos. De esta manera se conformaron los primeros equipos de IO los cuales contribuyeron a las mejoras de los radares, a la victoria de la guerra aérea que participó Gran Bretaña, manejo de operaciones antisubmarinas y de protección entre otras (Hillier & Lieberman, 2010). Finalizada la guerra y en vista de la evidente utilidad de la IO fue extendida en otros ámbitos de manera exitosa.

La IO tiene un gran rango de aplicación en la actualidad. Se puede distinguir algunas aplicaciones tipo como por ejemplo: asignación de recursos escaso, ordenamiento y coordinación de tareas (los primeros campos de aplicación de la IO se refería a este tipo de problemas), también existen los referentes a líneas de espera (este tipo de problemas se tienden sobre un mundo aleatorio normalmente en el cual se tratan las esperas y retrasos ocurridos en el sistema entre otros), por otro lado los de inventario, Costos y tiempo (se considera de las aplicaciones más sencillas relacionada a almacenar y/o mantener recursos; se estudia cuanto y cuando adquirir) y finalmente los de tipo gestión de proyectos (donde un determinado conjunto de tareas de un proyecto se modelan con la finalidad de conocer cuáles son los tiempos y tareas críticas (UROU, 2010).

En sí, se define a la IO como un enfoque científico para la toma de decisiones que buscan el mejor diseño y operar un sistema, normalmente en condiciones que requieren la asignación de recursos escasos (Winston, 2005).

2.2 MODELOS MATEMÁTICOS

Los modelos matemáticos o simplemente modelos¹ son empleados en la IO como las herramientas para lograr optimizar un determinado sistema es decir, lograr encontrar una alternativa de decisión con la propiedad de ser mejor que cualquier otra.

Estos modelos están compuesto de los siguientes elementos

- Función Objetivo. Es una medida cuantitativa del funcionamiento de un sistema que se desea minimizar o maximizar.

¹ El Modelo es un esquema teórico de un sistema o de una realidad compleja, por lo general expresado matemáticamente DRAE.



- Variables. Estas pueden ser ya sea independientes o dependientes según sea el caso y estas variables son las decisiones que afectan el valor de la función objetivo.
- Restricciones. Se definen como un conjunto de relaciones que las variables están obligadas a satisfacer de una u otra manera.

Estos tres componentes que conforman los modelos de IO tienen por finalidad determinar el valor que deben tomar las variables para lograr hacer óptima la función objetivo satisfaciendo el conjunto de restricciones planteadas, es decir resolver el modelo.

2.3 CONSTRUCCIÓN DE UN MODELO

De acuerdo con (Ramos & Begoña, 2012), en la construcción de un modelo matemático en un determinado sistema que se requiera atender mediante IO se distinguen siete faces.

2.3.1 Identificación, Descripción y Características del Problema. Es uno de los aspectos más importantes de la modelación, en este punto se efectúa una recolección de información relevante, se trata de definir el problema en términos vagos, se deberá prestar especial atención a la calidad de datos con los que se contará; esta fase es vital para que las decisiones sean útiles.

2.3.2 Especificación y Formulación Matemática. Se lleva a cabo la definición de las variables, ecuaciones, función objetivo, parámetros. En este punto se identifica el tipo de problema el cual puede ser de programación lineal, lineal entero mixto, cuadrático, no lineal, dinámica entre otros. Es importante lograr una relativa precisión en la formulación y llevar a cabo un análisis del tamaño y la estructura del problema.

2.3.3 Resolución. Construcción del algoritmo de obtención de solución la cual puede ser óptima, cuasi-óptima o por lo menos satisfactoria, dependiendo del tipo de problema se podrá efectuar diversos métodos de solución los cuales deberán ser correctamente abordados.



- Métodos Heurísticos. Aquellos que consiguen o bien, una solución aceptable de la función objetivo o en algunos casos, logran la mejor solución para la función.
- Métodos Óptimos. Aquellos que consiguen satisfacer las necesidades del modelo es decir, cumplen las restricciones solicitadas y con ello brindan la mejor solución a la función objetivo.

2.3.4 Verificación, validación y refinamiento. En esta fase se trata de comprobar el modelo construido tratando de eliminar los posibles errores en codificación así como lograr refinarlo en lo posible; comprobar la capacidad del modelo a ser adaptado al entorno real y determinar posibles ampliaciones del modelo a nuevas necesidades.

2.3.5 Interpretación y análisis de resultados. Esta es una fase crucial del modelo, para esto se requiere un conocimiento profundo a cerca del sistema que se está estudiando con la finalidad de interpretar lo que nos está indicando la solución del modelo. Hay que lograr un análisis de sensibilidad en parámetros de entrada y obviamente conseguir una robustez de la solución óptima.

2.3.6 Mejoras. Esta fase podrá o no darse, todo ello dependerá del análisis de datos, validación de resultados etc. del modelo en cuestión, la cual puede producir que se deba retornar a otras fases donde pudiera existir el problema. Esta fase se repetirá hasta lograr el objetivo para el cual el modelo ha sido diseñado.

2.3.7 Implantación, documentación y mantenimiento. Una vez finalizadas las fases anteriores de construcción de un modelo es necesario realizar una documentación clara, precisa y completa del modelo en cuestión; si es el caso realizar un manual de usuario con especificación, técnica, funcional, matemática e informática.

2.4 CUTTING AND PACKING PROBLEMS (C&P).

Los Cutting and Packing Problems o problemas de corte y empaquetado son una rama de la IO. Estos son entendidos como problemas combinatorios geométricos. Se puede encontrar una gran cantidad de variantes derivadas de la amplia gama de

aplicaciones prácticas y dependiendo de quién lo esté tratando (Salto, 2009). De acuerdo al número de dimensiones relevantes, los problemas de corte y empaquetado se pueden clasificar en unidimensionales, bidimensionales o tridimensionales.

En la siguiente figura se puede apreciar los tipos de corte.

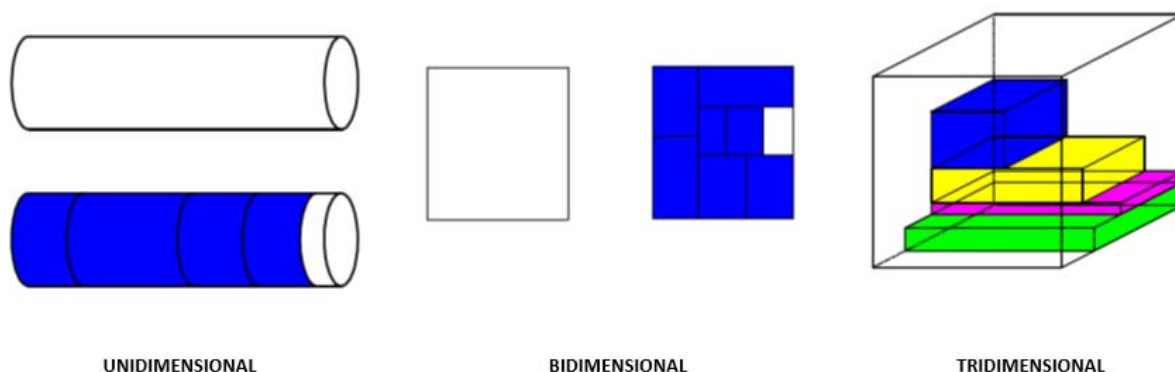


Ilustración 1: Tipos de Corte. Fuente: Autor

En la figura anterior se apreciar lo siguiente: En el corte unidimensional, se tiene un objeto tipo a ser cortado y luego del corte obtenemos un objeto cortado en 4 ítems en el cual también se tiene un desperdicio. En el corte bidimensional de un objeto tipo se supone una placa la cual será cortada y luego del corte se consiguen 7 ítems en el cual también se tiene un desperdicio. Finalmente, en el corte Tridimensional se supone una caja tipo en la cual se colocan o almacenaran 4 cajas de menor tamaño.

Existe una gran cantidad de problemas de Corte y Empaquetado (C&P) que se han introducido a lo largo de estos últimos años. A continuación se describen brevemente 3 problemas básicos y de los cuales parten o son la base para la construcción de nuevos problemas de C&P todo ello a partir de tipos de asignación y surtido de ítems pequeños.

2.4.1 Problema de la Mochila (Knapsack Problem). Este tipo de problemas trata de un objeto que tiene la particularidad de tener una capacidad fija el cual será llenado u ocupado por una serie de ítems con un determinado



tamaño menor al del objeto fijo de tal manera que se consiga un subconjunto de ítems que logren la máxima utilidad siempre y cuando satisfaga la restricción de la capacidad de la mochila (White, 1992).

2.4.2 Problema de la Caja de Embalaje (Bin Packing Problem). Este tipo de problemas tiene diversas aplicaciones como carga de vehículos o la planificación de rutas. Hace referencia al número mínimo de objetos (cajas) que se necesitan para empaquetar un determinado conjunto de ítems. Los objetos pueden ser modelados ya sea en una o más dimensiones (Coffman, et al., 1997).

2.4.3 Problema de Corte de Stock (Cutting Stock Problem). Este tipo de problemas tiene por finalidad satisfacer las demandas de unas determinadas piezas que se deben obtener mediante el corte de un objeto más grande disponible en stock. La distribución generalmente se realiza persiguiendo el objetivo de minimizar los cortes o piezas residuales generadas por el corte (Gracia Calandín, 2010).

2.5 CORTE DE STOCK (CSP)

De los problemas mencionados anteriormente, la investigación se centrara en un tipo específico, los Cutting Stock Problem o Problemas de Corte de Stock los cuales tiene por fin, determinar cómo debe cortarse un conjunto de ítems o piezas, las cuales tienen formas regulares o irregulares, todo ello a partir de formas más grandes las mismas que podrían ser también regulares o irregulares todo esto con el objetivo ya sea de maximizar la utilización o su equivalente, minimizar la pérdida de éste (Garcés, 2007).

Como se menciona anteriormente los problemas de corte y empaquetado pueden ser de orden unidimensionales, bidimensionales o tridimensionales. Para el estudio del caso propuesto, referente a cortes de varillas solamente se tendrá cortes transversales de estas a partir de una longitud estándar. En vista de esto se tiene un problema de corte unidimensional.

2.5.1 Problema de Corte Unidimensional (1D CSP)

Los problemas de corte unidimensional comúnmente denominados Problema de Corte Estándar Unidimensional (1D CSP) se encuentran definidos de la siguiente manera: m, L, l, d donde L representa la longitud del objeto en stock, m es el número



de ítems que se deberán producir, l sera la longitud de cada ítem y finalmente, d representa la demanda. Este tipo de modelos por lo general tiene por fin minimizar el número de objetos en stock que logren satisfacer la demanda o lo que es lo mismo minimizar las pérdidas de material por desperdicio (Gracia Calandín, 2010). Para el caso de estudio sobre corte de varillas de construcción se adopta a L como representación de la longitud de las varillas comerciales estándar, m viene a representar el número de tipos de barras que se requiere producir, l será la longitud de cada barra y finalmente, d representa igualmente la demanda.

2.5.1.1 Aplicaciones 1D CSP

Existen diversas aplicaciones de 1D CSP en la industria, se puede mencionar algunas como la Industria de la Madera en la cual se realizan los cortes estándar de listones de un mismo ancho utilizados para la construcción de diferentes muebles; la industria papelera con la finalidad de atender los varios solicitudes de los clientes que solicitan rollos de papel de diferente longitud; la industria de la construcción minimizando los desperdicios de las barras de diversos materiales empleados en las distintas construcciones, logrando reducir los costos; industrias de cable en la cual se puede tratar de reducir o minimizar el sobrante en el corte de rollos de cables y alambres (Solano Lazo & Ganoza Salazar, 2004) así como otras tantas industrias que tienen problemas de cortes de sus materias primas lo cual repercute en altos porcentajes de desperdicio y por ende provoca reducción de las ganancias esperadas y en otros casos perdidas de los recursos.

Existen diferentes formulaciones matemáticas con la finalidad de lograr modelar cortes unidimensionales

- a. Modelo de asignación
- b. Modelo basado en patrones de corte
- c. Modelo de corte único
- d. Modelo de grafos de flujo

a. Modelo de asignación

En 1939 se dio la primera formulación basada en variables de asignación en las cuales se empleaba variables binarias con la finalidad de relacionar las barras o ítems con el material que se disponía en stock. De esta manera se formuló el siguiente modelo (Kantorovich, 1960):

$$z = \min \sum_{i=1}^n y_j \quad (2.1)$$



Restricciones

$$\sum_{j=1}^n x_{ij} \geq d_i \quad i = 1, \dots, m \quad (2.2)$$

$$\sum_{i=1}^m l_i x_{ij} \leq L y_j \quad j = 1, \dots, n \quad (2.3)$$

$$y_j \in \{0, 1\} \quad j = 1, \dots, n \quad (2.4)$$

$$x_{ij} \geq 0 \text{ y entero, } i = 1, \dots, m \quad j = 1, \dots, n \quad (2.5)$$

Donde se define a y_j como variables binarias que representan a la elección de los objetos en stock j , a x_{ij} como el número de ítems demandados de longitud l_i asignados a un objeto en stock j . Donde las inecuaciones (2.2) y (2.3) representan respectivamente las condiciones de demanda y de la mochila². El Modelo de Asignación tiende a crecer rápidamente en tamaño a medida que se aumentan los valores de m y n . También puede llegar a presentar algunos problemas como son: una cota inferior muy pobre, puede conducir a soluciones diferentes en términos de los valores de las variables pero en la realidad son exactamente iguales.

b. Modelo basado en Patrones de Corte

Este modelo plantea que en lugar de hacer una asignación directa entre los ítems y los objetos en stock, se utiliza el concepto de patrón de corte. Un patrón de corte es una combinación factible de ítems para cada objeto en stock. Todo esto teniendo como premisa que la suma de las longitudes de los ítems producidos a partir de un objeto no llegue a sobrepasar la longitud del objeto. Un patrón j viene definido por el vector columna $a_j = (a_{1j}, \dots, a_{mj}) \in \text{enteros positivos}$, donde $j =$

² Condición de Mochila. La suma de las longitudes de las barras cortadas en un perfil no puede sobrepasar la longitud de este.



1, n. Se indica que es factible un patrón siempre y cuando cumpla la siguiente restricción (Gilmore & Gomory, 1963).

$$\sum_{i=1}^m l_i a_{ij} \leq L \quad j = 1, \dots, n \quad (2.6)$$

Cabe destacar que este modelo fue obtenido mediante la aplicación de la descomposición de Dantzing-Wolfe³.

c. Modelo de Corte Único

El modelo de corte único trabaja con la aplicación de cortes únicos con la finalidad de conseguir los ítems demandados. Este tipo de modelos atienden a la posibilidad de realizar un corte a uno de los objetos en stock, donde podría aparecer una pieza residual. Este residuo dependiendo de su tamaño podría tomarse ya sea como un scrap⁴ o podría reutilizarse si es el caso. Este modelo toma en cuenta el caso de múltiples longitudes en stock, donde el conjunto S representa las longitudes disponibles en stock para el corte, de esta manera $S = [W_1 \dots \dots W_k]$. Donde W representa las diversas longitudes disponibles en stock (Dyckhoff, 1981).

d. Modelo Grafos de Flujo

Esta formulación se basa en modelos de teoría de grafos⁵. En este modelo se tiene un grafo $G = (X, A)$ con $|X| = L + 1$ donde L es la longitud de los objetos en stock siendo todas iguales. Cada uno de los nodos existentes en el grafo X representan un posición dentro del objeto que va a ser cortado y cada uno de los arcos (h, j) el emplazamiento de un ítem de tamaño j h donde la posición h esta lo más a la izquierda posible del ítem. De esta manera definimos un patrón de corte como una secuencia de arcos que empiezan en el nodo. Por tanto, para un conjunto de arcos A y variables X_{hj} se formula lo siguiente (Gracia Calandín, 2010).

³ Se trata de un Algoritmo para resolver problemas de programación lineal. Este trabajo fue publicado en 1960 por George Dantzig y Wolfe Phil.

⁴ Definimos Scrap como un trozo sobrante de un corte que ya no podrá ser reutilizado. Es decir, un desperdicio.

⁵ La teoría de grafos se basa en la representación de un conjunto de puntos unido por líneas todo ello con la finalidad de estudiar las interrelaciones entre las unidades que se encuentran en interrelación. Muy utilizados en aplicaciones como rutas entre ciudades, tiempos máximos y mínimos en un proceso, flujo y control en un programa entre otros.



$$Z \text{ grafo} = \min Z \quad (2.7)$$

Restricciones

$$-\sum_{h,j \in A} X_{hj} + \sum_{j,l \in A} X_{jl} = \begin{cases} z & \text{si } j = 0 \\ 0 & \text{si } j = 1, \dots, L-1 \\ -z & \text{si } j = L \end{cases} \quad (2.8)$$

$$\sum_{h, h+li \in A} X_{h, h+li} \geq di \quad i = 1, \dots, m \quad (2.9)$$

$$X_{hj} \geq 0, \text{ entero} \quad (2.10)$$

Este modelo resulta equivalente al de patrones de corte con la debilidad de que el número de restricciones es un pseudo-polinomial⁶.

2.6 PROGRAMACIÓN LINEAL

La programación⁷ lineal es una herramienta de uso común en la actualidad que ha ayudado a ahorrar millones de dólares a muchas compañías; es tal su impacto que ha sido catalogada como uno de los avances científicos más importantes de mediados del siglo XX. Esta programación hace uso de un modelo matemático para describir el problema, en este tipo de problemas todas las funciones matemáticas empleadas en la modelación son funciones lineales. En fin, la programación lineal trata de la planeación de actividades con el fin de obtener un resultado óptimo (Hillier & Lieberman, 2010).

A continuación representamos matemáticamente un problema tipo de programación lineal

⁶ Denominamos pseudo-polinomial a los algoritmos donde la cota de tiempo de ejecución es polinomial en los enteros de la entrada y no sus logaritmos.

⁷ El termino programación no se refiere a términos computacionales; en esencia es sinónimo de planeación.



Función Objetivo

$$Z = c_1X_1 + c_2X_2 + \dots + c_nX_n \quad (2.11)$$

Restricciones

$$\begin{aligned}
a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n &\leq b_1 \\
\dots\dots\dots &\leq \dots \\
\dots\dots\dots &\leq \dots \\
a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n &\leq b_m
\end{aligned}
\quad (2.12)$$

Donde la función objetivo Z puede estar sujeta a minimización o maximización según sea el requerimiento del modelo y X_j representa las variables de decisión.

2.6.1 Programación Lineal Entera

Podemos definir a los modelos de programación entera como un modelo que contiene una función objetivo y restricciones idénticas a las formuladas por planeación lineal. Siendo la única diferencia que una o más variables de decisión tienen que tomar un valor entero para la solución final (Merino M., 2013). La forma simple de minimización es la siguiente:

$$\mathit{Min} [c] [x] \quad (2.13)$$

Sujeto a

$$[A][x] \geq [b] \quad (2.14)$$

$$x \in Z^+ \quad (2.15)$$



2.6.1.1 Clasificación de la Programación Entera

Existen tres tipos de modelos de programación entera

1. Pura. Como su nombre lo indica, es un problema en el cual absolutamente todas las variables de decisión adquieran valores enteros.
2. Mixta. En este tipo de problemas parte de las variables de decisión tienen valores enteros y la otra parte cumple con suposiciones de divisibilidad. Es decir, algunas de las variables son enteras, las otras no.
3. Binaria. En este tipo de problemas la solución de las variables de decisión tienen exclusivamente solo dos posibilidades de solución 0 ó 1.

El número de soluciones de un modelo de programación lineal entera a diferencia de problemas con variables reales es finito. Pero hay que tener en cuenta que a pesar de ello, el determinar todas las posibles soluciones en un problema de orden mediano podría resultar oneroso. En un problema con n variables debemos explorar 2^n soluciones (Castro, 2010).

A lo largo de los años se ha desarrollado diferentes métodos con la finalidad de lograr explorar de forma más eficiente las soluciones. Muchas de estas metodologías están apoyadas en la lógica del branch and bound (ramificación y poda) al cual nos referiremos más adelante.



CAPITULO III

CONSTRUCCIÓN DEL MODELO MATEMÁTICO

Con la finalidad de diseñar un programa que permita optimizar el corte de varillas de construcción es necesario construir un modelo matemático que permita lograr nuestro cometido. De esta manera se procede a realizar la modelación, la cual será la base para implementar la herramienta informática de optimización.



3.1 IDENTIFICACIÓN, DESCRIPCIÓN Y CARACTERÍSTICAS DEL PROBLEMA

Como se mencionó en el capítulo anterior dentro de la IO, existe una rama denominada problemas de corte y empaquetado, hace referencia a cortar un todo en un determinado número de piezas de menor dimensión con la finalidad de optimizar al máximo dichos cortes, logrando el menor desperdicio.

3.1.1 Planteamiento del Modelo

El modelo que se plantea con la finalidad de optimizar las varillas de construcción es de tipo corte unidimensional, la modelación atiende a un problema de programación lineal. Se tenga en cuenta que las variables de decisiones deberán ser enteras ya que obviamente no se pueden comprar un número fraccionario de varillas. En vista de esto se concluye que estamos ante un problema de Programación Lineal Entera Pura (PLEP).

El planteamiento del modelo se sustenta en el principio basado en patrones de corte presentado por (Gilmore & Gomory, 1963) *Un patrón de corte es una combinación factible de ítems para cada objeto en stock* (ver Capítulo II).

Con la finalidad de optimizar el corte de las varillas de construcción nos planteamos lo siguiente:

El problema trata de realizar cortes sobre los objetos en stock en el caso de estudio varillas de acero de construcción que cuentan con una determinada longitud estándar (en el mercado se tiene por lo general 6, 9 y 12 metros) con la finalidad de lograr los pedidos con el menor número de objetos. Por lo tanto, el objetivo es minimizar la cantidad de unidades de materia prima que satisfaga una determinada demanda. Para esto, determinar cómo cortar el menor número de longitudes comerciales L atendiendo una demanda di de piezas de tamaño Li de tal manera que el residuo Ri (cantidad sobrante de los cortes) sea mínima.

A continuación mostramos un ejemplo ilustrativo del problema a optimizar:

Vamos a suponer que tenemos un objeto en stock (varilla comercial) de longitud de 12 metros del cual se tiene un número infinito de objetos. A su vez se solicita que a partir de la varilla comercial se corten varillas obviamente de menor tamaño a la comercial de la siguiente manera: 1 varilla de 12 metros, 4 varillas de 5 metros, 4 varillas de 1 metro y finalmente 15 varillas de 3 metros.



Utilizando posibles patrones de corte a través de la operación matricial $[P][X] \geq [D]$ donde $[P]$ es la matriz de patrones, $[X]$ representa la periodicidad de cortes y $[D]$ la demanda que cada uno de los cortes planteados. De esta manera tenemos:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 \\ 0 & 4 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \geq \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \quad (3.1)$$

A continuación se ilustra los patrones de corte a manera de varillas, donde C representa a cada columna de la matriz $[P]$ de las cuales, las partes blancas de las varillas representan el desperdicio. De esta manera se tiene lo siguiente:

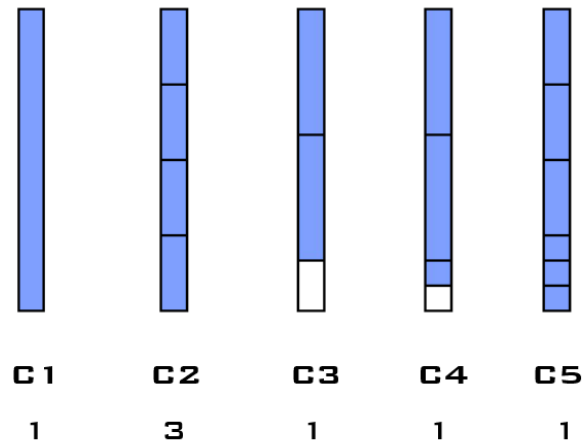


Ilustración 2: Esquemas de Patrones de Corte. Fuente: Autor

Solucionando la operación matricial tenemos que para satisfacer las demandas del problema se deberá utilizar un total de 7 varillas las cuales deberán ser cortadas, según los patrones C1, C3, C4, C5 una varilla para cada caso y 3 varillas para el patrón C2. Al finalizar el proceso se tendrá un desperdicio de 3 metros.

En resumen, la solución del modelo consiste en encontrar el valor de las variables de decisión X (patrón de corte), para los que la función objetivo alcance su valor mínimo donde el valor de las variables está sujeto a varias restricciones en nuestro caso demanda de las piezas.



3.1.2 Objetivos del Modelo.

El modelo planteado tiene por objetivo principal lograr minimizar los desperdicios por corte empleando el mínimo de materia prima que satisfaga las necesidades planteadas, es decir que se utilice el menor número de varillas comerciales para ser cortadas en secciones de menor tamaño según la necesidad.

3.1.3 Restricciones del Modelo

El modelo planteado deberá cumplir las siguientes restricciones.

- Al ser un problema de Programación Lineal Entera, Las Variables de Decisión deberán ser única y exclusivamente enteras.
- Las Variables de Decisión deberán ser positivas en decir, restringidas a la no negatividad.
- Restricción de la Mochila⁸.
- Restricción de Residuo. El sobrante o scrap de un determinado número de cortes sobre el objeto en stock (varillas estándar comerciales) deberá ser menor a la longitud del más pequeño de los cortes solicitados.
- Restricción de Demanda. El modelo deberá satisfacer los cortes solicitados

3.1.4 Función Objetivo

La función Objetivo estará conformada por las variables de decisión que hacen referencia al número de varillas que se deberán emplear para conseguir la solución óptima. La función estará enfocada hacia la minimización.

3.1.5 Variables decisión

Como se menciona anteriormente las variables de decisión deberán ser positivas y enteras, estas variables hacen referencia a la frecuencia de los patrones de corte, es decir cuántas veces se deberá repetir un determinado patrón con el fin de solucionar el modelo. Algunas de estas variables pueden adquirir valores de cero lo cual indica que ese determinado patrón al cual está referida esa variable no es factible.

⁸ En términos de peso, los objetos colocados en la mochila deben maximizar el valor sin exceder el peso máximo capas.



3.2 ESPECIFICACIÓN Y FORMULACIÓN MATEMÁTICA.

Para la formulación se procede de la idea de determinar cómo cortar el menor número de longitudes comerciales L atendiendo una demanda De de piezas de tamaño l_i de tal manera que el residuo r (cantidad sobrante de los cortes) sea mínima.

En vista de lo mencionado anteriormente se procede a formular el modelo matemático para optimizar el corte de varillas de construcción:

Variables de Decisión:

$j = Patrones$
 $i = Longitud de corte$

- $X_j =$ Número de varillas a cortarse a partir de un patrón j ($j = 1, 2, \dots, n$)

Función Objetivo:

$$\text{Min } Z = X_1 + X_2 + X_3 + \dots + X_n \quad (3.2)$$

Restricciones:

$$\begin{aligned} c_{11} * X_1 + c_{12} * X_2 + c_{13} * X_3 + \dots + c_{2n} * X_n &\geq De_1 \\ c_{21} * X_1 + c_{22} * X_2 + c_{23} * X_3 + \dots + c_{2n} * X_n &\geq De_1 \end{aligned} \quad (3.3)$$

.
. .
.

$$c_{m1} * X_1 + c_{m2} * X_2 + c_{m3} * X_3 + \dots + c_{mn} * X_n \geq De_m$$

$$X_1 + X_2 + X_3, \dots, X_n \geq 0, X_n \in Z^+ \quad (3.4)$$



Donde

- c_{ij} = Frecuencia o cantidad de cortes de una pieza de longitud i ($i = 1, 2, \dots, m$) dentro de una barra o esquema de corte según el patrón j ($j = 1, 2, \dots, n$).
- De_i = Demanda de piezas solicitadas según una longitud de corte i ($i = 1, 2, \dots, m$).

3.2.1 Restricciones de la Matriz de Patrones

La matriz de patrones deberá pasar por una especie de refinación de patrones antes de proceder a la solución de Modelo aplicado. De esta manera la matriz de patrones deberá sujetarse a las siguientes condiciones.

1. *Restricción Mochila.*

$$L - \sum_{i=1}^m c_{ij} * l_i \geq 0 \quad \forall j \quad (3.5)$$

2. *Restricción de patrones inútiles (Restricción del Residuo)*

$$L - \sum_{i=1}^m c_{ij} * l_i < l_{menor} \quad \forall j \quad (3.6)$$

Donde

- l_i longitud de corte demandado i ($i = 1, 2, \dots, m$).
- c_{ij} = Frecuencia o cantidad de cortes de una pieza de longitud i ($i = 1, 2, \dots, m$) dentro de una barra o esquema de corte según el patrón j ($j = 1, 2, \dots, n$).
- l_{menor} pieza que posee la longitud más corta de entre todas las demás piezas requeridas.
- L longitud del Objeto en stock (varilla comercial).



3.2.2 Patrones de Corte

Como se ha podido observar anteriormente, el modelo implementado para la optimización utiliza patrones de corte. En base a las piezas demandadas construimos los patrones de corte que son en sí, combinaciones geométricas de manera que la suma de las diversas partes en las que se divida una sección tipo no sobrepase la longitud del total de la sección.

De esta manera la matriz que estará constituida por los patrones de corte en función de la longitud estándar comercial L y las piezas demandadas l_i obedecerá a la siguiente expresión.

$$c_{1n}li_1 + c_{2n}li_2 + c_{3n}li_3 + \dots + c_{mn}li_m \leq L \quad (3.7)$$

Donde

- o l_i longitud de corte demandado i ($i = 1, 2, \dots, m$).
- o c_{ij} = Frecuencia o cantidad de cortes de una pieza de longitud i ($i = 1, 2, \dots, m$) dentro de una barra o esquema de corte según el patrón j ($j = 1, 2, \dots, n$).

3.2.2.1 Obtención de Patrones de Corte

Partimos de lo siguiente; un número x puede estar contenido un número de veces n en otro número L si cumple que $L > x$. De esta manera se determina cuántas piezas de longitud x máximas pueden ser cortadas a partir de una longitud estándar L . Con estos valores obtenidos procedemos a combinarlos unos con otros a manera de combinatoria basada en anclar y ramificar. A partir de los diversos valores obtenidos procedemos a aplicar las condiciones de mochila y de patrones inútiles indicados anteriormente, consiguiendo una posible matriz de patrones refinada.

Para un mayor entendimiento vamos a suponer que a partir de una varilla de longitud estándar comercial $L = 12 m$ necesitamos un número o demanda de piezas de $l_i = 3 m$ y $l_i = 8 m$. De esta manera la varilla comercial deberá ser cortada para satisfacer las demandas establecidas.

Ahora, para los dos pedidos l_i demandados y el L establecido se determina los esquemas posibles. Para $li = 8m$ se tiene $\langle 0 \ 1 \rangle$ posibilidades o esquemas y para $li = 3m$ se tiene $\langle 0 \ 1 \ 2 \ 3 \ 4 \rangle$ posibilidades o esquemas. Estos vectores indican las diferentes posibilidades que tiene cada l_i de estar contenidos en L .



A partir de estos valores se pueden generar los patrones de corte. De esta manera tenemos:

$$\begin{aligned} &pt\ 1 \begin{bmatrix} 0 \\ 0 \end{bmatrix}, pt\ 2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}, pt\ 3 \begin{bmatrix} 0 \\ 2 \end{bmatrix}, pt\ 4 \begin{bmatrix} 0 \\ 3 \end{bmatrix}, pt\ 5 \begin{bmatrix} 0 \\ 4 \end{bmatrix} \\ &pt\ 6 \begin{bmatrix} 1 \\ 0 \end{bmatrix}, pt\ 7 \begin{bmatrix} 1 \\ 1 \end{bmatrix}, pt\ 8 \begin{bmatrix} 1 \\ 2 \end{bmatrix}, pt\ 9 \begin{bmatrix} 1 \\ 3 \end{bmatrix}, pt\ 10 \begin{bmatrix} 1 \\ 4 \end{bmatrix} \end{aligned}$$

Ilustración 3 Posibles Patrones de Corte. Fuente: Autor

De acuerdo a la ilustración 3 existe un total de 10 posibles patrones de corte, la primera fila de cada vector hace referencia a los esquemas a partir de $l_i = 8m$ y la segunda fila hace referencia al esquema $l_i = 3m$. Ahora, sometemos a los patrones a las restricciones de refinamiento de la matriz de patrones.

- Aplicando la condición de la mochila, tenemos que los patrones (*pt*) 8, 9 y 10 no cumplen lo establecido ya que estos sobrepasan la cantidad de piezas que podría cortarse a partir de nuestro $L = 12m$ establecido. Por tanto, se eliminan como posibles patrones eficientes.
- Aplicando la condición de los patrones inútiles, los patrones (*pt*) 1, 2, 3, 4 y 6 no cumplen con lo requerido ya que estos cortes no satisfacen el valor del desperdicio mínimo para ser tomados como posibles patrones eficientes.

Luego de haber pasado los patrones a través de las restricciones de la matriz de patrones se definen cuales si pueden ser contados como posibles patrones de corte eficientes, estos son:

$$\begin{aligned} &pt\ 5 \begin{bmatrix} 0 \\ 4 \end{bmatrix} \text{ y } pt\ 7 \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &mppce = \begin{bmatrix} 0 & 1 \\ 4 & 1 \end{bmatrix} \end{aligned}$$

Ilustración 4: Posibles Patrones de Corte Eficiente. Fuente: Autor



De esta manera estos dos patrones constituyen lo que se denomina matriz de posibles patrones de corte eficiente [*mppce*].

3.2.3 Matriz de Posibles Patrones de Corte Eficiente [*mppce*]

El ejemplo expuesto anteriormente resulta muy escueto en comparación con las necesidades que nos podemos encontrar en el día a día en la construcción. En vista de esto resulta necesario implementar un algoritmo de codificación con la finalidad de conseguir nuestra [*mppce*] de manera óptima. Por ello en virtud de las posibilidades del software de programación matemática **MATLAB** en su versión **R2013** se ha procedido a realizar la programación con la finalidad de obtener dicha matriz que es la base para construir el programa de optimización de las varillas de acero de construcción **LCQ**.

3.3 RESOLUCIÓN DEL MODELO.

3.3.1 Método de Anclado y Ramificado para la Conformación de Patrones de Corte.

El algoritmo implementado para determinar [*mppce*] está conformado por varias rutinas que tiene como fundamento el anclado y ramificado con la finalidad de explorar los distintos patrones de corte que se pueden determinar a partir de la longitud comercial de la varilla a utilizar y las diferentes piezas demandadas.

La técnica de anclar y ramificar es un conjunto de rutinas que hace referencia a anclar en un determinado momento un determinado valor, fila o columna y hacer que varíen el resto de valores, filas o columnas a lo largo de su rango de maniobra lo cual se denomina ramificar.

Todos estos valores encontrados con la técnica de anclado y ramificado como se mostró en el ejemplo anterior son refinados es decir eliminar posibles soluciones deficientes caso de soluciones que no cumplan las restricciones o soluciones repetidas entre otras.

A continuación se muestra el diagrama de flujo de la técnica empleada. Se puede encontrar la codificación pertinente en el Anexo 1.

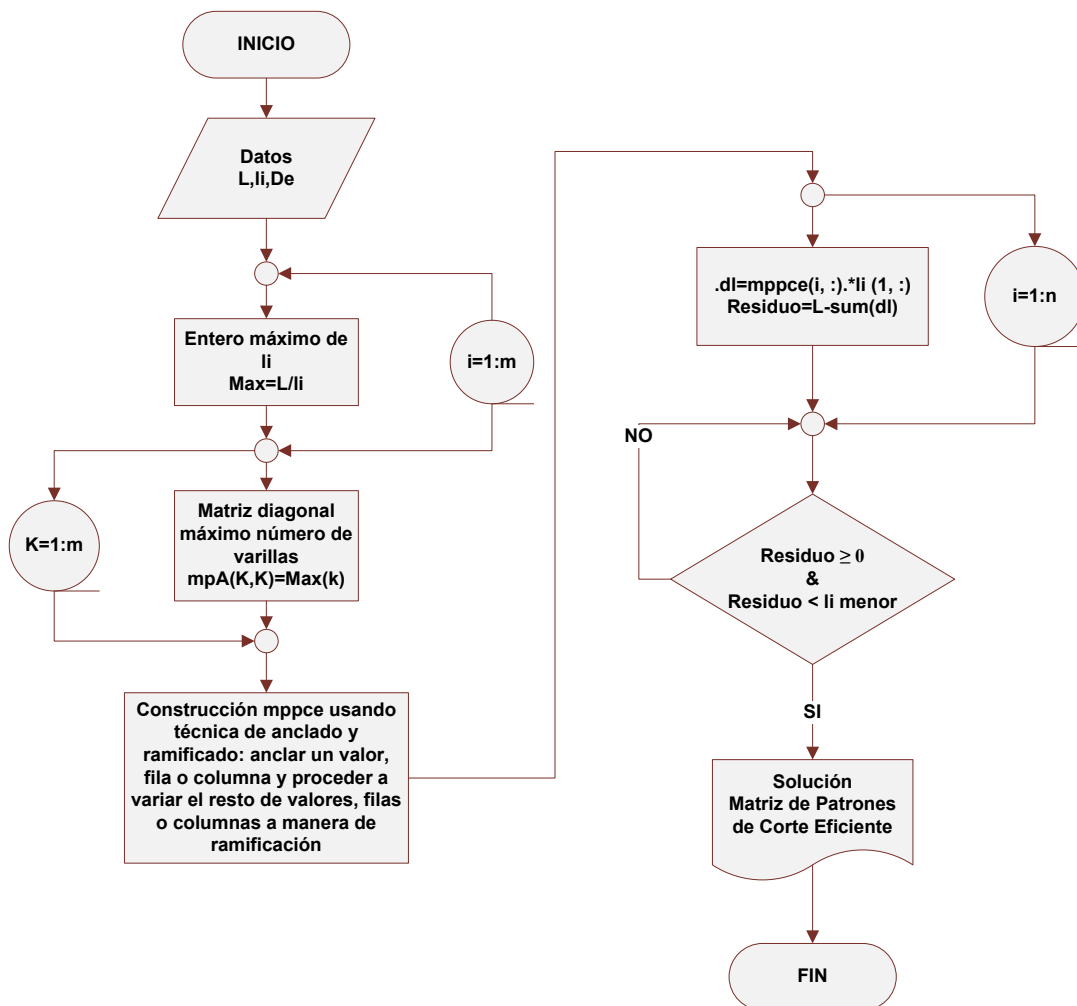


Ilustración 5 Diagrama del Algoritmo Solución de la Matriz de Posibles Patrones de Corte Eficiente. Fuente: Autor

3.3.2 Matriz de Patrones de Corte Eficiente

Una vez determinado $[mppce]$ se ha conformado todos nuestros datos para solucionar el modelo matemático que obedece al orden de la programación lineal entera pura como se indicó anteriormente.

De esta manera se tiene definido el modelo matemático a optimizar.

Variables de Decisión:

$j = \text{Patrones}$
 $i = \text{Longitud de corte}$

- X_j = Número de varillas a cortarse a partir de un patrón j ($j = 1, 2, \dots, n$)



Función Objetivo:

$$\text{Min } Z = X_1 + X_2 + X_3 + \dots + X_n \quad (3.8)$$

Restricciones

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1n} \\ c_{21} & c_{22} & c_{23} & \dots & c_{2n} \\ c_{31} & c_{32} & c_{33} & \dots & c_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & c_{m3} & \dots & c_{mn} \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_m \end{bmatrix} \geq \begin{bmatrix} De_1 \\ De_2 \\ De_3 \\ \vdots \\ De_m \end{bmatrix} \quad (3.9)$$

$$X_1 + X_2 + X_3 + \dots + X_n \geq 0, X_n \in Z^+ \quad (3.10)$$

Donde

- o c_{ij} = Frecuencia o cantidad de cortes de una pieza de longitud i ($i = 1, 2, \dots, m$) dentro de una barra o esquema de corte según el patrón j ($j = 1, 2, \dots, n$).
- o De_i = Demanda de piezas solicitadas según una longitud de corte i ($i = 1, 2, \dots, m$).

3.3.3 Optimización Mediante Matlab.

Dentro de la gran gama de herramientas (Toolbox) que presenta MATLAB⁹ el cual tiene por finalidad facilitar el trabajo al programador, se puede encontrar herramientas diseñadas para optimización, las cuales pueden ser utilizados para la solución de modelos matemáticos diseñados a través de la IO. Estas herramientas cuentan con distintas rutinas para optimización según sea la necesidad. En el caso de estudio existen funciones para solución de problemas de Programación Lineal.

⁹ Es un software matemático que cuenta con un entorno de desarrollo integrado (IDE) y un su propio lenguaje de programación (M). su nombre proviene de la abreviatura Matrix Laboratory (Laboratorio de matrices)



3.3.4 Función Linprog.

La función Linprog (linear programming) es empleado en la solución de problemas de programación lineal. El cual emplea algoritmos de resolución lineal continua fuertemente influenciada por el algoritmo Simplex¹⁰.

Este tipo de función obedece a los problemas del formato siguiente:

$$\text{Min } f'x \quad (3.11)$$

$$[A][x] \leq [b] \quad (3.12)$$

$$[Aeq][x] = [beq] \quad (3.13)$$

$$[Lb] \leq x \leq [Ub] \quad (3.14)$$

Donde

- $x \in \mathbf{R}^n$ las variables de decisión, $f \in \mathbf{R}^n$ coeficientes de la función objetivo.
- $[A][x] \leq [b]$ y $[Aeq][x] = [beq]$ sistema de desigualdades lineales y ecuaciones lineales en formato matricial. $[Lb]$ (lower bound) y $[Ub]$ (upper bound) son vectores \mathbf{R}^n en los cuales sus coordenadas coinciden con las cotas inferior y superior de las variables del problema (Canovas, et al., 2011).

3.3.4.1 Notación Linprog

- $x = \text{linprog}(f, A, b)$. Esta es la notación más simple de la función. Se emplea en resoluciones de corte lineal en donde sus restricciones atienden única y exclusivamente a desigualdades.
- $x = \text{linprog}(f, A, b, Aeq, beq)$. Solución de un problema discreto que satisface la igualdad de restricciones.

¹⁰ El método Simplex (George Dantzig, 1947). Es un método analítico de solución de Problemas de Programación Lineal en el cual reduce sistemáticamente un número infinito de soluciones a un número finito, el método utiliza el procedimiento de eliminación Gauss-Jordan.



- $x = \text{linprog} \langle f, A, b, \text{Aeq}, \text{beq}, \text{Lb}, \text{Ub} \rangle$. Solución de Modelos que presentan cotas inferiores y superiores respecto a la variable x .
- $x = \text{linprog} \langle f, A, b, \text{Aeq}, \text{beq}, \text{Lb}, \text{Ub}, x_0 \rangle$. Cuando se requiere indicar el punto de partida del algoritmo¹¹.
- $x = \text{linprog} \langle f, A, b, \text{Aeq}, \text{beq}, \text{Lb}, \text{Ub}, x_0, \text{options} \rangle$. Cuando se desea modificar determinados parámetros del proceso de resolución.

3.3.5 Optimización Linprog y Branch and Bound

Concretamente la función Linprog que forma parte del paquete de herramientas de **MATLAB** es la indicada para solucionar modelos matemáticos de programación lineal. Con la técnica de Linprog basada en Branch and Bound se puede ir examinando las posibles soluciones enteras hasta conseguir determinar la solución óptima del modelo matemático planteado. En sí, el método empleado contempla ir adicionando restricciones al problema hasta conseguir la solución entera solicitada en las variables de decisión.

El procedimiento Branch and Bound (Ramificación y Acotamiento) de acuerdo con (Castro, 2010) está fundamentado en la existencia de un número finito de soluciones posibles es decir, no todas factibles para un problema con enteros, los cuales se pueden representar mediante un diagrama de árbol. En el cual no es necesario enumerar todas las posibles soluciones si se puede eliminar algunas ramas. Teniendo en cuenta que es posible eliminar una rama si se consigue demostrar que no contiene una solución factible que sea mejor que una ya conseguida.

En fin, el método de Ramificación y Acotamiento es una especie de árbol de soluciones en el cual, cada una de las ramas del árbol nos podrá llevar a posibles soluciones. En este método el algoritmo ayuda a no despilfarrar recursos y procesos en soluciones que no son óptimas ya que se encarga de detectar en que ramificación las soluciones dadas ya no están siendo óptimas, de esta manera se procede a podar esa dicha rama del árbol de soluciones.

3.3.6 Solución Linprog y Branch and Bound

La solución trata en una parte inicial de satisfacer el problema original de manera que las variables enteras puedan tomar valores continuos¹² al cual se conoce como

¹¹ El defecto del algoritmo a larga escala y del algoritmo simplex es que ignoran los puntos de partida. Por tanto en ocasiones resulta necesario definir este valor.

¹² Los valores continuos son los valores (números reales) que se encuentran comprendidos en un rango determinado



A_0 . De esta manera, si se consigue que el problema tenga solución entera en todas sus variables, se ha obtenido la solución óptima.

Ahora, de no presentar el problema en cuestión la solución entera se dividirá el problema en dos subproblemas tomando una variable con valor actual no entero con el fin de separar, con lo cual se logra dos subproblemas.

- *Subproblema 1.* Es el problema A_0 mas una restricción

$$S_v \leq [S_v^\circ]$$

- *Subproblema 2.* Es el problema A_0 mas una restricción

$$S_v \leq [S_v^\circ] + 1$$

Donde

- $[S_v^\circ]$ es el mayor entero contenido en la variable S_v que es apartada.

A su vez, los subproblemas se tiene que resolver de la misma manera que el primero y de no encontrar la solución entera se debe repetir el proceso hasta agotar el espacio solución (Carreño, et al., 2012).

En sí, la solución del caso consiste en resolver el Problema Lineal Entero a través de la solución de un conjunto de problemas lineales que no son más que versiones relajadas del Problema Lineal Entero Mixto.

Es importante determinar si la solución es posible mediante una especie de sondaje de la siguiente manera:

- Tiene solución entera
- No tiene solución entera pero tiene una solución de menor calidad que la de la mejor solución entera ya determinada.
- Solución del problema no viable.

A continuación se muestra el diagrama de flujo de la técnica empleada para la solución del modelo matemático. Podemos encontrar la codificación pertinente en el Anexo 1.

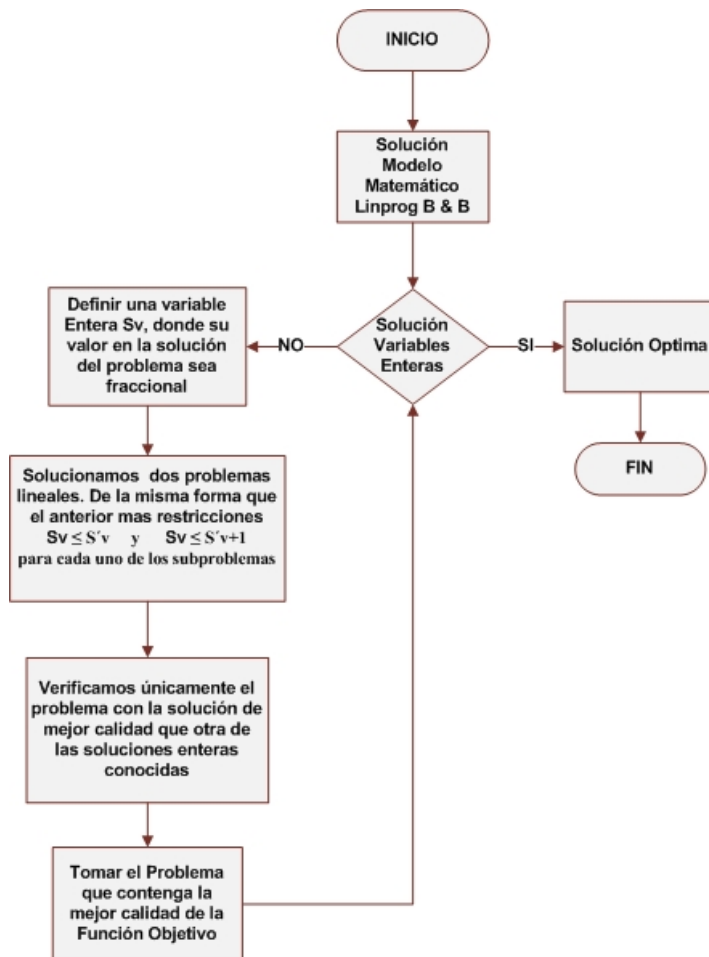


Ilustración 6: Diagrama del Algoritmo Solución del Modelo Matemático por Linprog B &B. Fuente: Autor



CAPITULO IV

IMPLEMENTACION Y RESULTADOS DE LA HERRAMIENTA INFORMATICA LCO

Una de las partes finales en el desarrollo de un modelo matemático una vez verificado su funcionamiento, notación, estructura, alcance, etc. se procede a la implementación del modelo mediante herramientas informáticas. A continuación se da a conocer el programa informático que se ha diseñado para la optimización de corte de varillas de acero al que se ha denominado “Linear Cutting Optimization” (LCO).

4.1 ESTRUCTURA DEL PROGRAMA

El programa diseñado LCO para optimización de corte de varillas de acero de construcción trabaja bajo el software MATLAB en su versión R2013 en el cual están implementados todos los algoritmos mencionados en el capítulo anterior. A continuación se muestra el diagrama de flujo de la herramienta LCO. Se puede encontrar la codificación pertinente en el Anexo 1.

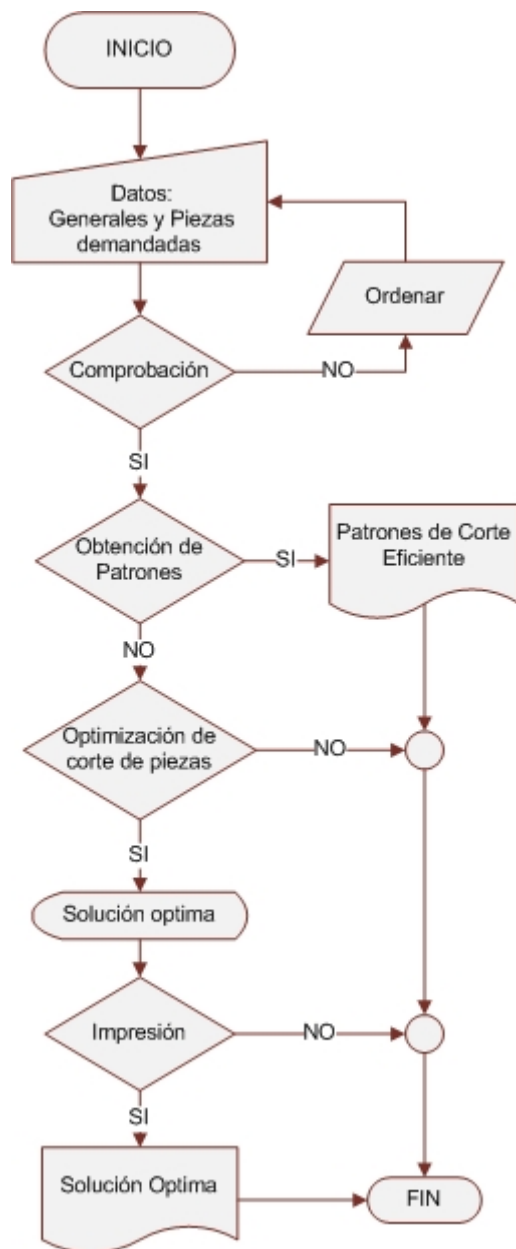


Ilustración 7: Diagrama de la herramienta informática LCO. Fuente: Autor



4.2 USO DE LCO

El programa LCO permite la optimización del corte de varillas de acero de construcción en el cual mediante la introducción de los diversos datos útiles para ejecutar el programa, esta nos dará a conocer la forma de obtener nuestras piezas demandadas a partir de las planillas de hierros de los planos estructurales de forma eficiente en términos de costo y desperdicio.

A continuación mediante capturas de imágenes del programa se muestra su funcionamiento.

4.2.1 Datos de entrada

Para la ejecución del programa LCO es necesario la introducción de datos. Estos datos están divididos en generales y piezas demandadas.

4.2.1.1 Generales.

Se indican los datos generales a ser introducidos de forma manual.

DATOS GENERALES			
PROYECTO	<input type="text" value="ICO"/>	DIAMETRO (mm)	<input type="text" value="14"/>
REALIZADO POR	<input type="text" value="Gustavo I. Ochoa R."/>	LONG. COMERCIAL(m)	<input type="text" value="6"/>
FECHA	<input type="text" value="01/06/2014"/>	PESO (kg/m)	<input type="text" value="1.21"/>
CONSTRUCTOR	<input type="text" value="Standard Construction"/>	COSTO (\$/kg)	<input type="text" value="1"/>

Ilustración 8: Datos generales a introducir en LCO

- ❖ Proyecto. Nombre del proyecto en el cual se desea aplicar la optimización de corte de varillas.
- ❖ Realizado por. Nombre de la persona o grupo encargado de la optimización.
- ❖ Fecha. fecha en la cual se ha realizado la optimización.
- ❖ Constructor. Contratista encargado del proyecto en cuestión.
- ❖ Diámetro. Introducción del diámetro en milímetros de la varilla que se desea optimizar.



- ❖ Longitud Comercial. Longitud en metros del objeto en stock es decir la longitud estándar comercial de las varillas que se desea adquirir.
- ❖ Peso. Peso en kilogramos por metro lineal respecto a la varilla comercial. Este dato se obtiene de forma automática al introducir el diámetro de la varilla comercial.
- ❖ Costo. Costo en el mercado del kilogramo de Acero.

4.2.1.2 Piezas Demandadas.

Los siguientes datos son obtenidos a partir de las planillas de hierros de los planos estructurales.

ENTRADA									
	Marca	Ubicación	Tipo	Doblado	a (m)	b (m)	c (m)	long. corte (m)	# de piezas
1	1000	PLINTO 800X370	Estribo	O	1	1	0.5000		7
2	1001	COLUMNA A1	Ganchos	C1	0.6000	0.3000	0.1000		7
3	1002	ZAPATA C2	Longitudinal	L	3	0.3000	NaN		10
4	1003	MURO M3	Longitudinal	I	6	NaN	NaN		15
5	1003	VIGA VC 450x250	Estribos	O	0.1900	0.3900	0.0800		20

Ilustración 9: Datos de Piezas Demandadas a introducir en LCO

- ❖ Marca. Marca a la cual atiende la pieza a optimizar.
- ❖ Ubicación. Lugar donde será utilizada dicha pieza (vigas, plintos, etc.)
- ❖ Tipo. Doblado que se efectuara a la pieza cortada (estribo, longitudinal, Gancho, etc.).
- ❖ Doblado. tipo de doblado. Aquí se deberá indicar cuál es la forma del doblado (I, O, L, C, C1).
- ❖ a, b, c. Dimensiones en metros de la pieza final.

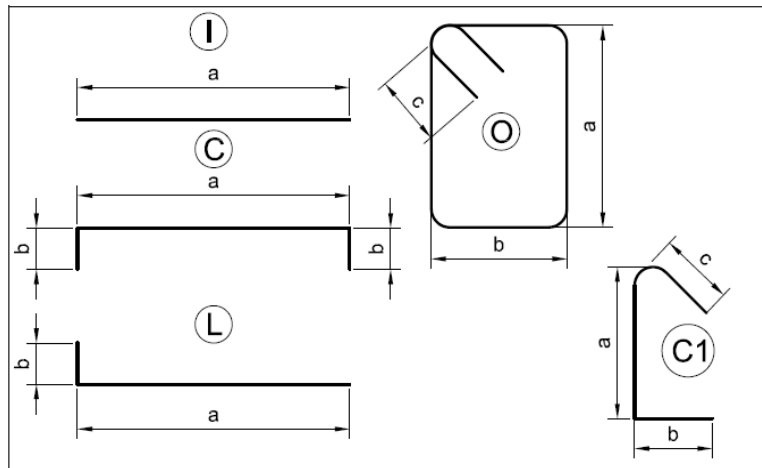


Ilustración 10: Tipos de Doblados de Hierros

- ❖ Longitud de Corte. Longitud total de la pieza a ser cortada. Este valor se obtiene de forma automática a partir de la introducción de las dimensiones de la pieza (a,b,c).
- ❖ Número de piezas. Demanda de piezas de una determinada longitud de corte.

4.2.1 Ejecutar LCO

La herramienta informática de optimización de corte lineal LCO brinda dos posibilidades, optimizar el corte de varillas u obtener los patrones de corte.

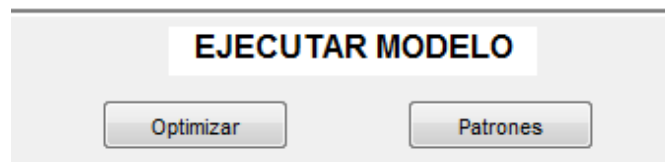


Ilustración 11: Soluciones que puede ejecutar LCO



4.2.2 Solución LCO

Una vez introducido los datos pertinentes al programa y solicitado la ejecución del mismo se logrará las soluciones de la optimización como se muestra en la siguiente figura.

SALIDA				
	Patrón	Esquema de Corte / Nro cortes x Long. Corte mtrs	Nro Varillas	Desperdicio x unid mtrs
1	1	1 x 6	15	0.00
2	2	1 x 3.3 \ 2 x 1.32	10	0.06
3	3	1 x 5 \ 1 x 1	7	0.00

Ilustración 12: Salida de resultados de la optimización en LCO

- ❖ Patrón. Patrón al que hace referencia un determinado esquema de corte. El número de patrones está directamente influenciado con el número de piezas solicitaras.
- ❖ Esquema de Corte. Esquema de corte que se deberá realizar. Por ejemplo 1x6 indica que se deberá realizar 1 corte de 6 metros.
- ❖ Nro. Varillas. Número de varillas de longitud estándar que serán necesarias para satisfacer las demandas impuestas y serán cortadas según un determinado esquema de corte.
- ❖ Desperdicio. Desperdicio producido por cada varilla según un determinado esquema.

Número Total de Varillas	32
Porcentaje Total de Desperdicio	0.31
COSTO TOTAL	232.32

Ilustración 13: Solución para satisfacer las necesidades logrando el menor desperdicio posible en LCO

- ❖ Número total de varillas. número total de barras de longitud comercial que se deberán adquirir para satisfacer las necesidades logrando el menor desperdicio posible.



- ❖ Porcentaje de Desperdicio. porcentaje final de desperdicio que se ha logrado a través de la optimización.
- ❖ Costo Total. costo total que conlleva la operación aplicando la optimización.

A continuación se muestra una imagen del interfaz final del programa LCO para optimización de corte de varillas de acero de construcción.


Optimizacion

Archivo

LINEAR CUTTING OPTIMIZATION

DATOS GENERALES

PROYECTO	ICO	DIAMETRO (mm)	14
REALIZADO POR	Gustavo I. Ochoa R.	LONG. COMERCIAL(m)	6
FECHA	01/06/2014	PESO (kg/m)	1.21
CONSTRUCTOR	Standard Construction	COSTO (\$/kg)	1



ENTRADA

	Marca	Ubicación	Tipo	Doblado	a (m)	b (m)	c (m)	Long. Corte (m)	# Piezas
1	1000	PLINTO 800X370	Estribo	O	1	1	0.5000	5	7
2	1001	COLUMNA A1	Ganchos	C1	0.6000	0.3000	0.1000	1.0000	7
3	1002	ZAPATA C2	Longitudinal	L	3	0.3000	NaN	3.3000	10
4	1003	MURO M3	Longitudinal	I	6	NaN	NaN	6	15
5	1003	VIGA VC 450x250	Estribos	O	0.1900	0.3900	0.0800	1.3200	20

SALIDA

	Patrón	Esquema de Corte / Nro cortes x Long. Corte mtrs	Nro Varillas	Desperdicio x unid mtrs
1	1	1 x 6	15	0.00
2	2	1 x 3.3 \ 2 x 1.32	10	0.06
3	3	1 x 5 \ 1 x 1	7	0.00

EJECUTAR MODELO

Optimizar Patrones

Número Total de Varillas	32
Porcentaje Total de Desperdicio	0.31
COSTO TOTAL	232.32

Ilustración 14: Interfaz de la herramienta informática LCO



4.3 RESULTADOS

Con la finalidad de llevar acabo la verificación y validación del programa de optimización de corte de varillas de acero de construcción *LCD* se tiene el objetivo de lograr aplicar el programa de optimización a un proyecto real.

Concretamente la construcción del subterráneo de una edificación compuesta estructuralmente por perfiles de acero y hormigón armado ubicada en la ciudad de Cuenca y llevada a cabo por una constructora local. Por razones de privacidad no se me ha autorizado dar a conocer el nombre concreto del proyecto analizado así como planos de la edificación.

Con la finalidad de implementar la herramienta informática de manera óptima. Se efectuó una evaluación tanto de los planos generales como los de detalles estructurales. A su vez, se realizó una revisión de las especificaciones técnicas requeridas por la edificación.

4.3.1 Piezas de Acero Demandadas

A partir de los planos estructurales determinamos las piezas demandadas. De esta maneta aseguramos que se han empleado un total de 4709 piezas solicitadas distribuidas de la siguiente manera:

- Diámetro de 8 mm. 2696 piezas solicitadas.
- Diámetro de 10 mm. 1423 piezas solicitadas.
- Diámetro de 14 mm. 365 piezas solicitadas.
- Diámetro de 16 mm. 170 piezas solicitadas.
- Diámetro de 20 mm. 55 piezas solicitadas.

Cabe recalcar que ciertas marcas indicadas en las planillas de los planos estructurales del proyecto a analizar no se han tenido en cuenta debido a que unas de ellas son traslapadas, las cuales no se indican en la planilla mientras que otras son de tipo longitudinal mayor o igual a la varilla comercial con lo cual no necesitan ser cortadas.

Las diversas piezas empleadas serán utilizadas en los elementos estructurales como son: plintos, vigas de cimentación, vigas de cimentación gradas, muros y zapatas.

Con la finalidad de comparar los desperdicios y por ende el costo que se produciría con una determinada varilla de longitud comercial en comparación a otras en nuestro mercado comúnmente la de 6 m y 12 m se ha utilizado un costo del kilogramo de acero de 1 \$ (USD).



A continuación se muestra un cuadro resumido elaborado a partir de la planilla de hierros original de los planos de detalles estructurales.

Marca	Cod	Ubicación	Tipo	Doblado	Diámetro mm	# varillas/Piezas	a (m)	b (m)	c (m)	Long. Corte (m)
60	2	Plintos 800X370	Estribos	O	10	45	0.33	0.76	0.08	2.34
62	2	Plintos 800X370	Estribos	O	10	35	0.76	0.13	0.08	1.95
63	2	Plintos 650X300	Estribos	O	10	33	0.26	0.61	0.08	1.90
66	2	Plintos 650X200	Estribos	O	10	210	0.16	0.61	0.08	1.70
68	2	Plintos 550X200	Estribos	O	10	150	0.16	0.51	0.08	1.50
95	2	Plintos 495X200	Estribos	O	10	70	0.16	0.46	0.08	1.39
61	2	Plintos 800X370	Estribos	O	10	105	0.33	0.28	0.08	1.38
58	1	Viga cimentacion	Estribos	O	8	80	0.19	0.39	0.08	1.31
59	1	Viga cimentacion	Estribos	O	8	172	0.19	0.39	0.08	1.31
106	1	Viga cimentacion gradas	Estribos	O	8	40	0.19	0.39	0.08	1.31
201	1	Muros y zapatas M1	Estribos	O	8	1212	0.15	0.15	0.08	0.75
201	1	Muros y zapatas M2	Estribos	O	8	476	0.15	0.15	0.08	0.75
80	3	Plintos 650X300	Ganchos	C1	10	56	0.61	0.16	0.08	0.85
64	3	Plintos 650X300	Ganchos	C1	10	75	0.26	0.16	0.08	0.50
98	4	Viga cimentacion	Ganchos	C1	8	80	0.19	0.08	0.08	0.34
107	4	Viga cimentacion gradas	Ganchos	C1	8	40	0.19	0.08	0.08	0.34
65	5	Plintos 650X300	Longitudinal	L	20	36	2.16	0.24		2.40
59	5	Plintos 800X370	Longitudinal	L	20	19	1.76	0.24		2.00
67	6	Plintos 650X200	Longitudinal	L	16	70	1.76	0.24		2.00
69	6	Plintos 550X200	Longitudinal	L	16	50	1.76	0.24		2.00
96	6	Plintos 495X200	Longitudinal	L	16	50	1.76	0.24		2.00
205	7	Muros y zapatas M1	Longitudinal	L	10	194	1.40	0.18		1.58
207	7	Muros y zapatas M1	Longitudinal	L	10	194	1.48	0.10		1.58
207	7	Muros y zapatas M2	Longitudinal	L	10	85	1.48	0.10		1.58
205	7	Muros y zapatas M2	Longitudinal	L	10	23	1.37	0.18		1.55
223	7	Muros y zapatas M3	Longitudinal	L	10	78	1.48	0.10		1.58
224	7	Muros y zapatas M3	Longitudinal	L	10	16	1.37	0.18		1.55
226	7	Muros y zapatas M3	Longitudinal	L	10	54	0.65	0.30		0.95
91	8	Muros y zapatas M4	Longitudinal	L	8	120	0.50	0.20		0.70
232	8	Muros y zapatas M5	Longitudinal	L	8	65	0.50	0.20		0.70
105	9	Viga cimentacion gradas	Longitudinal	C	14	30	0.90	0.15		1.20
204	9	Muros y zapatas M1	Longitudinal	C	14	296	1.78	0.13		2.03
209	9	Muros y zapatas M2	Longitudinal	C	14	39	2.15	0.13		2.40
203	10	Muros y zapatas M1	Longitudinal	C	8	206	1.78	0.13		2.03
208	10	Muros y zapatas M2	Longitudinal	C	8	204	2.15	0.13		2.40

Tabla 1: Lista de piezas de acero solicitadas. Fuente: Autor



4.3.2 Optimización mediante Programa LCO

Una vez obtenida la lista de piezas demandadas se procede a la optimización. Como se mencionó anteriormente se realizó el ingreso de datos agrupándolos en función del diámetro es decir, por cada corrida del programa se ingresó un solo tipo de diámetro.

4.3.2.1 Resultados mediante LCO

❖ Demanda de 2696 piezas de longitud variable de 8 mm de diámetro.

Long. Estandar	Peso ML	Costo Kg	Total Varillas	Peso Total	Desperdicio	Costo total
L m	Kg	\$	#	Kg	%	\$
6	0.39	\$ 1.00	468	1095.12	3.00%	\$ 1.095.12
12	0.39	\$ 1.00	229	1071.72	0.78%	\$ 1.071.72

Tabla 2: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 8mm mediante LCO

De esta manera se observa que resulta más económico adquirir varillas de 12 m en comparación con las de 6 m para satisfacer la necesidad. Con esto se tendrá un desperdicio del 0.78% versus el 3.00% de desperdicio con la varilla de 6 m.

Por tanto, se procede a determinar los patrones de corte óptimo a partir de la varilla de 12m.

Cod	1,4,8,10
Diametro	8
Tipo	Estribo
Uso	Vigas de cimentacion, Muros y zapatas
L	12
# varillas	229

Patron	Esquema de Corte	Nro Varillas	Deperdicio x unidad mtrs
	Nro cortes X Long. Corte mtrs		
1	5x2.4	35	0.00
2	9x1.31	32	0.21
3	16x0.75	100	0.00
4	4x2.4/3x0.75	1	0.15
5	3x2.4/6x0.75	6	0.30
6	4x2.4/3x0.7	1	0.30
7	2x2.4/10x0.7	1	0.20
8	5x2.03/5x0.34	24	0.15
9	1x2.4/1x2.03/1x0.75/2x0.7/4x1.31	1	0.18
10	14x0.75	1	1.50
11	17x0.7	10	0.10
12	5x2.03/2x0.75	17	0.35
Desperdicio Total			0.78%

Tabla 3: Esquemas de corte óptimo para piezas de 8mm a partir de una longitud comercial de 12m mediante LCO



❖ Demanda de 1423 piezas de longitud variable de 10 mm de diámetro.

Long. Estandar	Peso ML	Costo Kg	Total Varillas	Peso Total	Desperdicio	Costo total
L	Kg	\$	#	Kg	m	\$
6	0.62	\$ 1.00	378	1406.16	6.48%	\$ 1.406.16
12	0.62	\$ 1.00	180	1339.20	1.48%	\$ 1.339.20

Tabla 4: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y 12m de 10 mm mediante LCO

De esta manera se observa que resulta más económico adquirir varillas de 12 m en comparación con las de 6 m para satisfacer la demanda. Con esto se tendrá un desperdicio del 1.48% versus el 6.48% de desperdicio con la varilla de 6 m.

De esta manera se procede a determinar los patrones de corte óptimo a partir de la varilla de 12m.

Cod	2, 3, 7
Diametro	10
Tipo	Estribo, Ganchos y Longitudinal
Uso	Plintos, Muros y zapatas
L m	12
# varillas	180

Patron	Esquema de Corte	Nro Varillas	Deperdicio x unid mtrs
	Nro cortes X Long. Corte mtrs		
1	6x1.95	5	0.30
2	6x1.9	2	0.60
3	7x1.7	30	0.10
4	8x1.5	18	0.00
5	8x1.39	1	0.88
6	8x1.38	2	0.96
7	1x2.34/5x1.9	4	0.16
8	1x2.34/6x1.5	1	0.66
9	1x2.34/7x1.38	12	0.00
10	4x2.34/1x1.9	1	0.74
11	2x2.34/5x1.39	11	0.37
12	2x2.34/5x1.38	1	0.42
13	5x1.95/1x0.95	1	1.30
14	7x1.39/2x0.95	1	0.37
15	7x1.55/1x0.95	3	0.20
16	5x1.58/4x0.95	5	0.30
17	7x1.58/1x0.85	52	0.09
18	7x1.58/1x0.5	7	0.44
19	6x1.58/5x0.5	12	0.02
20	5x1.58/1x1.55/1x0.95/1x0.85/1x0.5	4	0.25
21	7x1.55/2x0.5	2	0.15
22	12x0.95	2	0.60
22	7x1.58	3	0.94
Desperdicio Total			1.48%

Tabla 5: Esquemas de corte óptimo para piezas de 10mm a partir de una longitud comercial de 12m mediante LCO



❖ Demanda de 365 piezas de longitud variable de 14 mm de diámetro.

Long. Estandar	Peso ML	Costo Kg	Total Varillas	Peso Total	Desperdicio	Costo total
L	Kg	\$	#	Kg	m	\$
6	1.21	\$ 1.00	155	1125.3	27.18%	\$ 1.125.30
12	1.21	\$ 1.00	67	972.84	9.14%	\$ 972.84

Tabla 6: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 14 mm mediante LCO

De esta manera se observa que resulta más económico adquirir varillas de 12 m en comparación con las de 6 m para satisfacer la demanda. Con esto se tendrá un desperdicio del 9.14% versus el 27.18% de desperdicio con la varilla de 6 m.

Por tanto, se procede a determinar los patrones de corte óptimo a partir de la varilla de 12m.

Cod 9
Diametro 14
Tipo Longitudinal
Uso Vigas cimentacion gradas, Muros y Zapatas
L m 12
varillas 67

Patron	Esquema de Corte		Nro Varillas	Deperdicio x unid mtrs
	Nro cortes	Long. Corte mtrs		
1	5	2.03/1X1.2	30	0.65
2	2	2.4/3X2.03	6	1.11
3	1	2.4/4X2.03	27	1.48
4		5X2.03	4	1.85
			Desperdicio Total	9.14%

Tabla 7: Esquemas de corte óptimo para piezas de 14mm a partir de una longitud comercial de 12m mediante LCO



❖ Demanda de 170 piezas de longitud variable de 16 mm de diámetro.

Long. Estandar	Peso Unidad	Costo unidad	Total Varillas	Peso Total	Desperdicio	Costo total
L	Kg	\$	#	Kg	m	\$
6	1.58	\$ 1.00	57	540.36	0.58%	\$ 540.36
12	1.58	\$ 1.00	29	549.84	2.3%	\$ 549.84

Tabla 8: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 16 mm mediante LCO

Al contrario de los anteriores casos se observa que resulta más económico adquirir varillas de 6 m en comparación con las de 12 m para satisfacer la demanda. Con esto se tendrá un desperdicio del 0.58% versus el 2.30% de desperdicio con la varilla de 12 m.

De esta manera se procede a determinar los patrones de corte óptimo a partir de la varilla de 6m.

Cod 6
Diametro 16
Tipo Longitudinal
Uso Plintos
L m 6
varillas 57

Patron	Esquema de Corte		Nro Varillas	Deperdicio x unid mtrs
	Nro cortes	Long. Corte mtrs		
1	3x2		56	0.00
2	2x2		1	2.00
			Desperdicio Total	0.58%

Tabla 9: Esquemas de corte óptimo para piezas de 16mm a partir de una longitud comercial de 6m mediante LCO



❖ Demanda de 55 piezas de longitud variable de 20 mm de diámetro.

Long. Estandar	Peso ML	Costo Kg	Total Varillas	Peso Total	Desperdicio	Costo total
L	Kg	\$	#	Kg	m	\$
6	2.47	\$ 1.00	22	326.04	5.75%	\$ 326.04
12	2.47	\$ 1.00	11	326.04	5.75%	\$ 326.04

Tabla 10: Análisis costo desperdicio de la operación mediante longitudes comerciales de 6m y12m de 20 mm mediante LCO

En este caso resulta que utilizar varillas de 6 m o 12 m conlleva a una inversión igual y por ende un desperdicio similar de 5.75%.

De esta manera se procede a determinar los patrones de corte óptimo a partir de la varilla de 12 m.

Cod 5
Diametro 20
Tipo Longitudinal
Uso Plintos
Lm 12
varillas 11

Patron	Esquema de Corte		Nro Varillas	Deperdicio x unid mtrs
	Nro cortes	Long. Corte mtrs		
1	5	2	1	2.00
2	1	4	1	1.60
3	4	1	8	0.40
4	3	2	1	0.80
			Desperdicio Total	5.75%

Tabla 11: Esquemas de corte óptimo para piezas de 20mm a partir de una longitud comercial de 12m mediante LCO



CAPITULO V

CONCLUSIONES

A continuación se dará a conocer las diversas conclusiones a las que se a llegando finalizado la presente investigación.

5.1 Referente a la Modelación

- *Diseñar un modelo matemático es implementar lo mejor posible la realidad a través de ciertas fórmulas.*
- *Resulta imprescindible asegurarse de que el modelo represente adecuadamente la realidad que pretende reflejar.*



- *Tener en cuenta que el mejor modelo no sirve de nada si los datos de entrada no están adecuadamente refinados.*
- *La técnica de modelado, así como el nivel de detalle del mismo y en general cualquier aspecto de modelado debe adecuarse a la estructura del problema, a las expectativas y necesidades del cliente.*
- *El tiempo de menos dedicado al modelo conceptual retrasa de forma exponencial la implantación del modelo.*
- *El contraste de los resultados del modelo con los datos reales del sistema es imprescindible.*

5.2 Referente al Programa LCO

- *Se ha podido constatar la herramienta LCO permite optimizar el corte de varillas de acero de construcción lo cual repercute en una reducción del costo e impacto ambiental.*
- *Mediante LCO podemos comparar costos que conllevaría utilizar una determinada varilla comercial, esto permitirá escoger la mejor opción.*
- *la herramienta LCO permite determinar los esquemas óptimos de corte los cuales podrán ser entregados directamente al personal encargado del corte de las barras para que siga esos patrones y con ello disminuir el desperdicio y por ende el costo.*
- *Es necesario tener instalado una versión del software MATLAB en el computador para poder mover el programa.*
- *El nivel de porcentaje de desperdicio del programa LCO que se aplique a un determinado caso estará fuertemente influenciado por la cantidad de piezas solicitadas. A mayor número de piezas solicitadas menor el porcentaje de desperdicio mediante la herramienta LCO*



La herramienta informática *LCQ* presenta las siguientes limitaciones.

- *Tiempo de solución. Al programa LCQ puede tomarle desde décimas de segundo hasta tardar horas para mostrar la solución esto es debido a la cantidad de soluciones posibles que debe ir explorando hasta encontrar la solución óptima como se indicó en capítulos anteriores. Por tanto entre mayor sea el número de piezas demandadas mayor será el número de horas que tarde el programa en dar solución. Esto también estará influenciado en la capacidad del computador que este destinado a mover el programa. En vista de esto se ha optado por dar la posibilidad al usuario de obtener únicamente los posibles patrones de corte eficiente y con ello dar la posibilidad de utilizar un software especializado en problemas lineales como es el caso de CPLEX¹³.*
- *A pesar de ser un programa de optimización de corte lineal, está construido exclusivamente para problemas relacionados con solicitudes de ítems que responden a varillas de acero para construcción de elementos estructurales por ello no se recomienda utilizar el programa para otros tipos de ítems que no sean estos.*

5.3 Referente a la Investigación de Operaciones

- *Se ha escrito y se ha dicho tanto sobre esta rama que resulta difícil decir algo nuevo. Sin duda, la aplicación oportuna de la IO resulta vital en los diversos campos de la Ingeniería Civil. Teniendo en cuenta que cada vez los recursos son más limitados por ello estos deben ser aprovechados de forma óptima. En fin, podemos decir que la IO es sin duda la ciencia de la toma de decisiones.*

¹³ CPLEX es un solucionador de alto rendimiento que proporciona algoritmos matemáticos robustos que permite resolver problemas con millones de variables y restricciones ya sean de programación lineales, entera mixta o cuadrática.



5.3 Referente a los Objetivos Planteados.

- *Los objetivos planteados al inicio de esta investigación los cuales eran formular el modelo matemático a optimizar, diseñar un algoritmo de programación para el modelo y todo esto con el fin de construir un programa que permita optimizar el corte de varillas de acero en la construcción han sido alcanzados exitosamente los mismos que fueron desarrollados a lo largo del presente documento.*

En Fin, Esta investigación no ha hecho más que sembrar en mí la profunda sensación de lo mucho que hay que hacer aún por el desarrollo la Ingeniería Civil. No debemos pensar que ya todo está hecho o que existen formas, métodos establecidos para llevar acabo un determinado fin. Siempre debemos preguntarnos:

¿Habrá una mejor forma de hacerlo?.....



BIBLIOGRAFIA

- Arenas, F., 2008. Los Materiales de Construcción y el Medio Ambiente. *Revista*, 1(1), pp. 1-2.
- Canovas, M. J., Huertas, V. & Sempere, M., 2011. Prácticas de Optimización con Restricciones . En: ECU, ed. *Optimización Matemática Aplicada* . Alicante-España: Club Universitario, pp. 147-149.
- Carreño, E., Escobar, A. & Gallego, R., 2012. ALGORITMO DE BRANCH AND BOUND ESPECIALIZADO APLICADO AL PLANEAMIENTO DE SISTEMAS DE TRANSMISIÓN.. *Grupo de Investigación en Planeamiento de Sistemas Eléctricos*, 1(1), pp. 2-4.
- Castro, Y., 2010. *SISTEMATIZACIÓN DE DETALLES, HABILITACIÓN Y ARMADO DE ACEROS ASTM A615 PARA CONSTRUCCIONES DE CONCRETO ARMADO: Impacto Técnico, Económico y Ambiental*. Ayacucho - Peru: Universidad Nacional San Cristobal de Huamanga.
- Coffman, E. G., Johnson, D. S., Shor, P. W. & Weber, R. R., 1997. Bin Packing with discrete item sizes, part 2. *Random Structures and Algorithms*, 10(1), pp. 69 - 101.



- Diario El Universo, 2014. Industria de la Construcción, la que más apórtó al crecimiento del 2013. *El Universo*, 23 abril, pp. 1-2.
- Dyckhoff, H., 1981. A New Linear Programming Approach to the Cutting Stock Problem. *Operations Research* , 29(6), pp. 1092 - 1104.
- Garcés, J. E., 2007. Problemas de Corte de Piezas. En: *Optimización de los Esquemas de Corte en Aserraderos Forestales Mediante Algoritmos Metaheurísticos*. Concepción - Chile: s.n., pp. 16-18.
- Gilmore, P. C. & Gomory, R. E., 1963. A linear Programming Approach to the Cutting Stock Problem. *Operations Research* , 11(6), pp. 863 - 888.
- Gracia Calandín, C., 2010. Los Problemas de Corte y Empaquetado. En: *Metodos y Algoritmos para Resolver Problemas de Corte Unidimensional en Entornos Realistas*. Valencia - España: s.n., pp. 37 - 43.
- Hillier, F. S. & Lieberman, G. J., 2010. Introducción, Programación Entera. En: *Introducción a la Investigación de Operaciones*. Novena ed. s.l.:McGraw Hill, pp. 1-2,21,429-480.
- Kantorovich, L. V., 1960. Mathematical Methods of Organizing and Planning Production. *Management Science* , 6(4), pp. 366 - 422.
- Merino M., M., 2013. *Técnicas Clásicas de Optimización*. Valencia - España: Universidad Politécnica de Valencia.
- PHPSimplex, 2006. *PHPSimplex*. [En línea]
Available at: <http://www.phpsimplex.com/>
[Último acceso: 2014].
- Ramos, A. & Begoña, V., 2012. *Modelos Matemáticos de Optimización*. Madrid: Universidad Pontificia Comillas .
- Salto, C., 2009. En: *Metaheurísticas Híbridas Paralelas Para Problemas Industriales de Corte, Empaquetado y Otros Relacionados*. San Luis, Argentina: s.n., p. 49 55.
- Soibelman, L., 2010. Desperdicios vs el control de los materiales. *Instituto Mexicano del Cemento y del Concreto*, I(1), p. 3.
- Solano Lazo, U. C. & Ganoza Salazar, D., 2004. *Un Algoritmo de Búsqueda Adaptativa Aleatoria y Golosa para la Resolución del Problema de Cortes*, Lima - Peru: Universidad Nacional Mayor de San Marcos.
- UROU, 2010. Problemas Tipo en la Investigación de Operaciones. En: *Introducción a la Investigación Operativa*. Montevideo - Uruguay: Universidad de la Republica Oriental del Uruguay, pp. 7-8.



White, D. J., 1992. A Complementary greedy heuristic for the Knapsack problem..
European Journal of Operational Research, 62(1), p. 85.

Winston, W. L., 2005. Introducción a la Construcción de Modelos, Introducción a la Programación Lineal, Programación Entera. En: Thomson, ed. *Investigación de Operaciones Aplicaciones y Algoritmos*. Cuarta ed. Indianápolis(Indiana): Thomson, pp. 1-9,49-63,475-512.



ANEXOS

1 Código fuente de LCO en lenguaje Matlab

1.1 Función principal para optimización de corte de varillas de acero de construcción.

```
Function[mpsolucionT,TSC,vreptp,porcentajedesperdicio]=Linear_cutting_optimization(li,L,De,int)
```

1.2 Obtención de la matriz de Posibles Patrones de corte eficiente [mppce].

```
tic
%% (i). Datos de entrada
% L: longitud comercial de la varilla, usamos una de 9m.
% li:vector fila, longitudes demandadas a ser cortadas.
% De: vector fila, Numero de Varillas demanadadas por seccion.

%% (ii). Ordenar datos.
% ordenar matriz li, forma descendente.
li=sort(li,'descend');
%% (iii). Cortes maximos
% max: vector fila compuesto por el maximo numero de varillas
```




```
% que pueden cortarse apartir de L, respecto al pedido li.
max=floor(L./li);

%% (iv). matriz de patrones A
% dmax: matriz diagonal formada por el maximo numero de varillas, max
dmax=zeros(length(max));
for k=1:length(max)
    dmax(k,k)=max(k);
end
d=dmax;
mpA=dmax;
% end mpA

%% (v), matriz de patrones B
[fl cl]=size(dmax);
w=0;
for i=1:fl
    for j=i:cl
        for u=1:max(1,j)
            d(i,j)=[u];
            dl=d(i,:).*li(1,:);
            residuo=L-sum(dl);
            if residuo>=0
                w=w+1;
                mpB(w,:)=d(i,:);
            end
        end
    end
end
mpB;
% end mpB

%% (Vi), matriz de patronez C F & G
D=mpB;
[fl cl]=size(mpB);
w=0;
for i=1:fl
    for j=1:cl
        D(1,:)=mpB(i,:);
        for u=1:max(1,j)
            D(i,j)=[u];
            w=w+1;
            mpCFG(w,:)=D(i,:);
        end
    end
end
mpCFG;
% matriz de patronez C
w=0;
[fl cl]=size(mpCFG);
for i=1:fl
    dl=mpCFG(i,:).*li(1,:);
    residuo=L-sum(dl);
    if residuo>=0
        w=w+1;
        mpC(w,:)=mpCFG(i,:);
    end
end
mpC;
% end mpC.

% matiz de patrones F
% SOLUCION 1 HACER CEROS COLUMNAS CORRIDAS.
AAAA=mpCFG;
CCCC=mpCFG;
[fl cl]=size(mpCFG);
n=cl-1;
for i=1:n
    CCCC(:,i)=zeros;
    FFFF=[AAAA;CCCC];
end
```



```
AAAA=FFFF;
end
FFFF;

w=0;
[f1 cl]=size(FFFF);
for i=1:f1
    dl=FFFF(i,:).*li(1,:);
    residuo=L-sum(dl);
    if residuo>=0
        w=w+1;
        mpF(w,:)=FFFF(i,:);
    end
end
mpF;
% end mpF.

% matriz de patrones G
% SOLUCION 2 HACER CEROS COLUMNAS SALTADAS.
YYYY=mpCFG;
EEEE=mpCFG;
[f1 cl]=size(mpCFG);
n=cl;
for i=1:n
    EEEE(:,i)=zeros;
    GGGG=[YYYY;EEEE];
    YYYY=GGGG;
    EEEE=mpCFG;
end
GGGG;

w=0;
[f1 cl]=size(GGGG);
for i=1:f1
    dl=GGGG(i,:).*li(1,:);
    residuo=L-sum(dl);
    if residuo>=0
        w=w+1;
        mpG(w,:)=GGGG(i,:);
    end
end
mpG;
% end mp C F & G.

%% (vii), matriz de patrones D
[f1 cl]=size(dmax);
w=0;
mpD=[];
for m=1:f1-1
    for n=m+1:cl
        d(1,:)=dmax(m,:);
        for u=1:max(1,n)
            d(1,n)=[u];
            dl=d(1,:).*li(1,:);
            residuo=L-sum(dl);
            if residuo>=0
                w=w+1;
                mpD(w,:)=d(1,:);
            end
        end
    end
end
mpD;
% end mpD

%% (viii), matriz de patrones E H & I
unos=ones(1,length(max));
w=0;
mpEHI=[];
[f1 cl]=size(unos);
for j=1:cl
```



```
Do(1,:)=unos(1,:);
  for u=0:max(1,j)
    Do(1,j)=[u];
    w=w+1;
    mpEHI(w,:)=Do(1,:);

  end
end
mpEHI;

% matriz de patronez E
w=0;
[f1 cl]=size(mpEHI);
mpE=[]
  for i=1:f1
    dl=mpEHI(i,:).*li(1,:);
    residuo=L-sum(dl);
    if residuo>=0
      w=w+1;
      mpE(w,:)=mpEHI(i,:);
    end
  end
end
mpE;
% end mpE.

% matiz de patrones H
% SOLUCION 1 HACER CEROS COLUMNAS CORRIDAS.
AAAAA=mpEHI;
CCCCC=mpEHI;
[f1 cl]=size(mpEHI);
n=c1-1;
  for i=1:n
    CCCCC(:,i)=zeros;
    HHHHH=[AAAAA;CCCCC];
    AAAAA=HHHHH;
  end
end
HHHHH;

w=0;
[f1 cl]=size(HHHHH);
  for i=1:f1
    dl=HHHHH(i,:).*li(1,:);
    residuo=L-sum(dl);
    if residuo>=0
      w=w+1;
      mpH(w,:)=HHHHH(i,:);
    end
  end
end
mpH;
% end mpH.

% matriz de patrones G
% SOLUCION 2 HACER CEROS COLUMNAS SALTADAS.
YYYYY=mpEHI;
EEEEEE=mpEHI;
[f1 cl]=size(mpEHI);
n=c1;
  for i=1:n
    EEEEE(:,i)=zeros;
    IIIII=[YYYYY;EEEEEE];
    YYYYY=IIIII;
    EEEEE=mpEHI;
  end
end
IIIII;

w=0;
mpI=[]
[f1 cl]=size(IIIII);
  for i=1:f1
    dl=IIIII(i,:).*li(1,:);
```



```
residuo=L-sum(dl);
    if residuo>=0
        w=w+1;
        mpI(w,:)=IIIII(i,:);
    end
end
mpI;
% end mp E H & I.

%% (ix), Fusion de mps
Fusionmps=[mpA;mpB;mpC;mpF;mpG;mpD;mpE;mpH;mpI]
% end Fusionmps
%% (x), Refinar Fusionmps
% eliminar filas repetidas.
operador=Fusionmps;
ceros=Fusionmps;
[fl cl]=size(Fusionmps);
% paso 1, llevar a cero las filas repetidas a excepci3n la original.
for i=1:fl
    w=0;
    for j=1:cl
        if operador(i,:)==ceros(j,:)
            w=w+1;
            if w>=2
                ceros(j,:)=zeros(1,cl);
            end
        end
    end
end
end
ceros

% paso 2 eliminamos las filas con elementos nulos.
Refinarfusionmps=ceros(find(sum(ceros')>0),:);
Refinarfusionmps
% end Refinar Fusion mps

%% (xi), Matriz Solucion
% el menor valor de la demanda di
minimo=min(li);
li;
Refinarfusionmps;
al=[];
[fl cl]=size(Refinarfusionmps);
for i=1:fl
    al(i,:)=Refinarfusionmps(i,:).*li;
end
al;
%verificar desperdicio
w=0;
for i=1:fl
    residuo=L-sum(al(i,:));
    if residuo<minimo
        w=w+1;
        mpsolucion(w,:)=Refinarfusionmps(i,:);
    end
end
mpsolucion
%end mpsolucion
mpsolucionT=mpsolucion'
%end mpsolucionT
% end Programa patrones de corte

% finalizacion si se quiere solo patrones de corte
if int==1
    TSC=0;
    vreprtp=0;
    porcentajedesperdicio=0;
    toc
    return
end
```



1.3 Solución del Modelo Matemático

```
%% solucion modelo matematico
%Datos iniciales
%Matriz patrones
A=mpsolucionT.*-1;
%Vector Columna # varillas Demandas
b=(De.*-1)';
%Variables de Decision, Vector Columna
[fl cl]=size(mpsolucionT);
y=ones(1,cl);
f=y';
% otros parametros
Aeq=[];
beq=[];
%Cota inferior
lb=zeros(1,cl);
%Cota Superior
ub=Inf*(f);
%Vector de indices que apunta a las variables enteras
M=[1:length(f)];
%Tolerancia de la Integralidad
e=1*10^-9;
% La investigación incorpora éste código que pertenece a Sherif A.
%Tawfik, Faculty of Engineering, Cairo University.
%llamar a solucion modelo
[x, val, status]=Branchandbound(f, A, b, Aeq, beq, lb, ub, M, e);
```

1.4 Linprog y Branch and Bound (El presente código de la función “Branchandbound” pertenece a Sherif A. Tawfik, Faculty of Engineering, Cairo University para Problemas Lineales Enteros Puros).

```
function [x, val, status]=Branchandbound(f, A, b, Aeq, beq, lb, ub, M, e)
options = optimset('display', 'on');
bound=inf;
[x0, val0]=linprog(f, A, b, Aeq, beq, lb, ub, [], options);
[x, val, status, b]=rec(f, A, b, Aeq, beq, lb, ub, x0, val0, M, e, bound);
function [xx, val, status, bb]=rec(f, A, b, Aeq, beq, lb, ub, x, v, M, e, bound)
options = optimset('display', 'off');
[x0, val0, status0]=linprog(f, A, b, Aeq, beq, lb, ub, [], options);
if status0<=0 | val0 > bound
xx=x; val=v; status=status0; bb=bound;
return;
end
ind=find( abs(x0(M)-round(x0(M)))>e );
if isempty(ind)
status=1;
if val0 < bound
x0(M)=round(x0(M));
xx=x0;
val=val0;
bb=val0;
else
xx=x;
val=v;
bb=bound;
end
return
end
br_var=M(ind(1));
br_value=x(br_var);
if isempty(A)
[r c]=size(Aeq);
```



```
else
[r c]=size(A);
end

A1=[A ; zeros(1,c)];
A1(end,br_var)=1;
b1=[b;floor(br_value)];
A2=[A ;zeros(1,c)];
A2(end,br_var)=-1;
b2=[b;-ceil(br_value)];
[x1,vall,status1,bound1]=rec(f,A1,b1,Aeq,beq,lb,ub,x0,val0,M,e,bound);
status=status1;
if status1 >0 & bound1<bound
xx=x1;
val=vall;
bound=bound1;
bb=bound1;
else
xx=x0;
val=val0;
bb=bound;
end
[x2,val2,status2,bound2]=rec(f,A2,b2,Aeq,beq,lb,ub,x0,val0,M,e,bound);
if status2 >0 & bound2<bound
status=status2;
xx=x2;
val=val2;
bb=bound2;
end
```

1.5 Solución final: Patrones de corte eficientes, Porcentaje de Desperdicio, Barras a emplear.

```
%% Desperdicios
mpFinal matriz patrones final optimizada LOS QUE SE VAN A USAR
VariablesFO=x'
nc=numel(VariablesFO(VariablesFO==0));
[fl cl]=size(VariablesFO);
i=0;
for j=1:cl
i=i+1;
%ELIMINANDO COLUMNAS QUE NO VAN DE MPSOLUCIONT VariablesFO
if VariablesFO(1,i)==0
mpsolucionT(:,i)=[];
VariablesFO(:,i)=[];
i=i-1;
end
end
mpFinal=mpsolucionT
VariablesFOF=VariablesFO
%varras
numvarillas=sum(VariablesFO)
% Matriz tipo mT
[fl cl]=size(mpFinal);
mT=zeros(fl,numvarillas);
% Matriz mpEXT
variablesxxx=VariablesFOF;
mpEXT=[];
kkk=0;
aaa=1;
for j=1:cl
for i=aaa:numvarillas
mT(:,i)=mpFinal(:,j);
end
aaa=variablesxxx(j)+kkk+1;
kkk=variablesxxx(j)+kkk;
```



```
end
mpEXT=mT
%% Eliminacion de cortes innecesarios
mpCF=mpEXT;
% mp es mpExt
[f1 cl]=size(mpCF);
summ=[];
for i=1:f1
summ(i,1)=sum(mpCF(i,:));

end
summ;
Det=De';
for i=1:f1
if summ(i,1)>Det(i,1)
% el valor de i indica que fila hay que ajustar
i;
for o=1:cl
for z=1: mpCF(i,o)
if sum(mpCF(i,:))>Det(i,1)
mpCF(i,o)=mpCF(i,o)-1;
end
end
end
end
end
mpCF
for i=1:f1
summ2(i,1)=sum(mpCF(i,:));
end
summ2;
% Verificacion, comprobacion demanda De vs obtenida
compDevsObt=[Det summ2]
% MATRIZ a*li
li=li'
[f1 cl]=size(li);
[f c]=size(mpCF);
for i=1:f1
for j=1:c
mdesp(i,j)=li(i,1)*mpCF(i,j);
end
end
mdesp
%suma
for i=1:c
vsuma(1,i)=sum(mdesp(:,i));
end
vsuma
%desperdicio
Desperdicio=sum(L-vsuma)
porcentajedesperdicio=(Desperdicio*100)/(numvarillas*L)
%% Tabla soluciones
mpCFT=mpCF';
% eliminar filas repetidas.
operadorr=mpCFT;
ceross=mpCFT;
[f1 cl]=size(mpCFT);
% l vector de repitencia de patrones
zxx=1;
for i=1:f1
w=0;
if zxx<f1+1
for j=1:f1
if operadorr(zxx,:)==ceross(j,:)
w=w+1;

end
end
end
end
vreptp(i)=w;
zxx=zxx+w;
```



```
end
vrep_tpv;
% eliminar los elementos ceros del vector
vrep_tpv(vrep_tpv==0)=[]
% 2 llevar a cero las filas repetidas a excepción de la original.
for i=1:f1
    w=0;
    for j=1:f1
        if operador(i,:)==ceross(j,:)
            w=w+1;
            if w>=2
                ceross(j,:)=zeros(1,c1);
            end
        end
    end
end

end
ceross;
% eliminamos las filas con elementos nulos.
TSC=ceross(find(sum(ceross')>0),:);
TSC=TSC'
toc
```