



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



Facultad de Contaduría  
Administración e Informática

**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS**

**FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E  
INFORMÁTICA**

**DOS ENFOQUES METAHEURÍSTICOS PARA RESOLVER EL  
PROBLEMA DE IMPRESIÓN DE ETIQUETAS**

**TESIS  
PARA OBTENER EL GRADO DE:  
MAESTRÍA EN OPTIMIZACIÓN Y CÓMPUTO APLICADO**

PRESENTA:  
**L.I. ROCIO DIEGO CELIS**

DIRECTOR DE TESIS:  
DR. FEDERICO ALONSO PECINA

CO-DIRECTOR:  
DR. MARTÍN HERIBERTO CRUZ ROSALES

**CUERNAVACA, MORELOS**

**SEPTIEMBRE, 2019**

---

## AGRADECIMIENTOS

---

Agradezco a:

Dios por darme la vida, la sabiduría y fortaleza para poder realizar este trabajo de investigación.

A mi querida familia: mamá Rebeca, hermanos: Noemi, Eve, Candi, Inocencio, Oralia, Yesenia, Yasmin, Josue y Edey, por sus oraciones y apoyo incondicional.

A mis sobrín@s por su tierno cariño.

A mi asesor Federico Alonso Pecina, por su paciencia, su enseñanza y dirección para poder comenzar y terminar la maestría.

A los doctores José Crispin Zavala Díaz, José Alberto Hernández Aguilar, Martín Heriberto Cruz Rosales y Luis Manuel Gaggero Sager, por su apoyo, enseñanza, y por formar parte del jurado revisor del trabajo de investigación.

A todos los doctores que forman parte del cuerpo académico de la maestría en optimización y computo aplicado, por su enseñanza.

A mis compañeras y compañeros de la maestría que amablemente me brindaron su apoyo.

A toda mi familia, amigos y hermanos en Cristo, por su amistad y sus oraciones.

*Porque yo se los planes que tengo para ustedes, dice Jehová, planes de paz y no de mal, para darles un futuro y una esperanza. Jeremías 29:11*

---

## RESUMEN

---

En el presente trabajo de investigación de tesis se aborda el problema de impresión de etiquetas que se presenta en una imprenta, es un problema de optimización combinatoria y pertenece a la clase de complejidad NP-duro. Dada la dificultad práctica para resolver de forma exacta toda una serie de importantes problemas combinatorios, para los cuales es necesario ofrecer alguna solución debido a su interés práctico, se comenzaron a desarrollar algoritmos que proporcionen soluciones factibles, los cuales no garantizan encontrar el óptimo, pero si encuentran soluciones de buena calidad en un tiempo de cálculo aceptable. De acuerdo con la literatura, los algoritmos que se han utilizado para resolver este tipo de problema y que han dado buenos resultados son: Recocido Simulado, GRASP, Algoritmos Basados en Inmunidad y Búsqueda Tabú. En la presente tesis de maestría, se presentan dos enfoques metaheurísticos, los cuales son Aceptación por Umbral y Búsqueda Tabú, para resolver el problema de impresión de etiquetas. Los resultados obtenidos al implementar los algoritmos de Aceptación por Umbral y Búsqueda Tabú, muestran que se puede resolver el problema de impresión, obteniendo soluciones que mejoran a los trabajos reportados en el estado del arte en la mayoría de las instancias, en tiempos razonables.

---

## Abstract

---

This research work of thesis addresses the problem printing label that are presented in a printing press, it is a problem of combinatorial optimization and belongs to the complexity class NP-hard. Given the practical difficulty of solving an exact series of important combinatorial problems, for which it is necessary to offer some solution due to their practical interest, they who researches began to develop algorithms that provide feasible solutions, which do not guarantee to find the optimum, find solutions of good quality in an acceptable calculation time According to the literature, the algorithms that have been used to solve this type of problem and have given good results are: Simulated Annealing, GRASP, Algorithms Based on Immunity and Tabu Search. In the present thesis of mastery, two metaheuristics approaches are presented which are Threshold Accepting and Tabu Search, to solve the problem of label printing. The results obtained by implementing the Threshold Accepting and Tabu Search algorithms show that the printing problem can be solved by obtaining solutions that improve the work reported in the state of the art in most instances at reasonable times.

# Contenido

## Índice de tablas

## Índice de Figuras

## Glosario

CAPITULO 1. INTRODUCCIÓN .....	10
1.1 Antecedentes .....	10
1.2 Planteamiento del problema.....	11
1.3 Objetivos .....	12
1.3.1 General .....	12
1.3.2 Específicos .....	12
1.4 Alcances y limitaciones .....	12
1.5 Hipótesis.....	13
1.6 Justificación.....	13
1.7 Organización de la tesis .....	13
CAPÍTULO 2. MARCO TEÓRICO.....	15
2.1 Literatura del problema de impresión .....	15
2.2 Clasificación del problema de impresión .....	22
2.2.1 Problema de impresión de portadas sin costo de cliché .....	22
2.2.2 Problema de impresión de etiquetas con restricción de repetición .....	23
2.2.3 Problema de impresión de portada con costo de cliché.....	29
2.2.4 Problema de impresión de etiquetas sin restricción de repetición .....	29
CAPITULO 3. METAHEURÍSTICAS UTILIZADAS.....	30
3.1 Aceptación por Umbral.....	33
3.2 Búsqueda Tabú .....	35
CAPITULO 4. METODOLOGÍA.....	40
4.2 Aceptación por Umbral.....	49
4.3 Sintonización de parámetros del algoritmo de Aceptación por Umbral .....	53
4.4 Búsqueda local con doble vecindad .....	54
4.5 Búsqueda Tabú .....	55
4.6 Sintonización de parámetros del algoritmo de Búsqueda Tabú .....	57
CAPITULO 5. EXPERIMENTACIÓN Y RESULTADOS.....	59

CAPITULO 6. CONCLUSIONES Y TRABAJOS FUTUROS .....	65
6.1 Conclusiones.....	65
6.2 Trabajos futuros .....	65
REFERENCIAS .....	66

## Índice de tablas

Tabla 2. 1 Desglose para calcular función objetivo .....	26
Tabla 2. 2 Espacio de soluciones de instancia LPP18 .....	28
Tabla 4. 1 Cálculo de porcentaje de ejemplo de solución inicial .....	48
Tabla 4. 2 Configuración inicial .....	50
Tabla 4. 3 Seleccionando etiquetas a intercambiarse .....	50
Tabla 4. 4 Verificación de etiquetas .....	50
Tabla 4. 5 Nueva configuración.....	50
Tabla 4. 6 Configuración inicial .....	51
Tabla 4. 7 Verificando que parrillas sean diferentes.....	51
Tabla 4. 8 Nueva configuración.....	52
Tabla 4. 9 Resultados de sintonización de $\alpha$ y $u_{\min}$ .....	53
Tabla 4. 10 Resultados de sintonización de ciclo interno .....	54
Tabla 4. 11 Resultados de sintonización de los límites inferior y superior.....	57
Tabla 4. 12 Resultados de sintonización de ciclo externo .....	58
Tabla 5. 1 Resultados del algoritmo Aceptación por Umbral.....	59
Tabla 5. 2 Resultados del algoritmo Aceptación por Umbral en las seis instancias generadas .....	60
Tabla 5. 3 Resultados del algoritmo Búsqueda Tabú.....	61
Tabla 5. 4 Resultados del algoritmo Búsqueda Tabú en las seis instancias generadas .....	61
Tabla 5. 5 Comparación de mejores resultados de los algoritmos de Aceptación por Umbral, Búsqueda Tabú y el mejor del estado del arte .....	62
Tabla 5. 6 Comparación de mejores resultados de los algoritmos de Aceptación por Umbral, Búsqueda Tabú y el mejor de la literatura , del total de instancias ....	64

## Índice de Figuras

Figura 2. 1 Representación tipo A .....	15
Figura 2. 2 Representación tipo B .....	16
Figura 2. 3 Representación tipo C.....	16
Figura 2. 4 Representación de restricciones .....	25
Figura 3. 1 Clasificación general de los métodos de optimización mono-objetivo.	31
Figura 4. 1 Representación de asignación aleatoria de etiquetas a parrillas .....	43
Figura 4. 2 Representación cuando se agregan las etiquetas a las parrillas .....	44
Figura 4. 3 Identificación de la etiqueta con mayor demanda .....	44
Figura 4. 4 Resultado de la división entre la demanda de cada etiqueta y el número de asignaciones en la parrilla .....	45
Figura 4. 5 Identificación de etiqueta con mayor demanda, asignación de etiqueta y resultado de división .....	45
Figura 4. 6 Identificación de etiqueta con mayor demanda, asignación de etiqueta y resultado de división .....	46
Figura 4. 7 Identificación de número de veces que se imprime la parrilla uno .....	46
Figura 4. 8 Configuración total de la parrilla dos .....	47
Figura 4. 9 Configuración total de parrilla tres.....	47
Figura 4. 10 Diagrama general de algoritmo Aceptación por Umbral.....	52

## Glosario

**Metaheurística:** Es un algoritmo diseñado para resolver aproximadamente una amplia gama de problemas de optimización difíciles. La metaheurística se aplica generalmente a problemas para los cuales no existe un algoritmo satisfactorio específico para resolverlos. (BoussaïD et. al, 2013).

**Problemas de optimización combinatoria:** Son aquellos problemas de optimización donde las variables de decisión son enteras, y por lo general, el espacio de soluciones está formado por ordenaciones o subconjuntos de números naturales (Díaz et. al, 2000).

**Problema NP-duro:** Son problemas conocidos como NP-hard, para los que se ha conjeturado que no hay una constante  $n > 1$  para la cual el tiempo de solución esté limitado por un polinomio de grado  $n$ , (Siarry, 2016).

**Clase NP:** Corresponde a los problemas de decisión que tienen un sistema de prueba eficiente.

**Función objetivo:** La función que nos permite identificar la calidad de la solución, generalmente es maximizar o minimizar una expresión lineal o no lineal. (Díaz et. al, 2000).

**Parrilla:** Es una parte de una imprenta a gran escala. En ella se colocan los moldes de impresión de etiquetas, portadas, entre otros.

**Etiqueta:** Rótulo que se coloca adherido, enganchado, fijado o colgado en alguna parte de diferentes productos para indicar una marca, clasificación, precio, u otro tipo de información.

**Desperdicio:** Son aquellos materiales que son desechados. En las industrias representan una pérdida de dinero y recursos, debido a la ineficiencia de una máquina o el uso de dinero exagerado que salga de los presupuestos acordados para la producción.

# CAPITULO 1. INTRODUCCIÓN

## 1.1 Antecedentes

En la vida real existe gran cantidad de problemas prácticos, para los cuales no se conoce algún algoritmo eficiente que encuentre una solución óptima.

La teoría de la complejidad estudia el crecimiento del costo computacional, principalmente en memoria y tiempo. Dicha teoría tiene medidas de como aumenta el costo computacional al solucionar un problema respecto al tamaño del mismo. Algunas de las medidas son: lineal, logarítmica, cuadrática, polinomial, exponencial, entre otras. Los problemas que se pueden resolver en un tiempo polinómico se encuentran en la clase NP (Bratley, 1997).

En el presente trabajo de investigación se pretende buscar una solución satisfactoria para el problema de impresión en una imprenta, dicho problema es de optimización combinatoria, perteneciente a la clase de complejidad NP-duro (Ekici et. al, 2010).

La imprenta tiene un papel importante para la historia de la cultura universal. Es un método mecánico que consiste en reproducir textos e imágenes sobre papel, tela u otros materiales. La historia moderna sería inimaginable sin la invención de la imprenta ya que su uso cambió por completo la cultura e historia del mundo. Uno de los trabajos importantes que se realiza en la imprenta, es la impresión de etiquetas.

La impresión de etiquetas tiene muchas aplicaciones en la industria. En el sector de fabricación, existen grandes demandas y diferentes modelos de etiquetas. En la imprenta, las parrillas se configuran antes de que comience el proceso de impresión. El número de etiquetas que se pueden asignar a cada parrilla es fijo debido al tamaño de la misma. Se tiene que determinar el número de parrillas necesarias y la composición correspondiente de cada una de ellas. Un método simple que muchas imprentas a menudo adoptan es configurar la menor cantidad posible de parrillas. Dado que cada parrilla solo puede acomodar un número fijo de etiquetas, una tarea importante es elaborar las composiciones de las parrillas mediante la asignación adecuada de etiquetas, para cumplir con los requisitos del pedido de manera efectiva. La composición de cada parrilla es bastante arbitraria, lo que generalmente

termina con una cantidad excesiva de etiquetas impresas. Una buena composición de la parrilla mejorará significativamente la eficiencia del proceso de impresión y, al mismo tiempo, reducirá el desperdicio de recursos (Yiu et. al, 2007).

## 1.2 Planteamiento del problema

El problema de impresión tiene dos variantes principales, las cuales son: problema de impresión de portadas (cover printing problem) y problema de impresión de etiquetas (label printing problem) (Tuyttens & Vandaele, 2014).

El objetivo del problema de impresión de portadas, es minimizar el costo total de producción; las variables con las que trabaja esta variante son: número de portadas, demanda de cada portada, capacidad de cada parrilla, costo de configuración y costo de impresión.

Mientras que el objetivo del problema de impresión de etiquetas, es minimizar el porcentaje de desperdicio; las variables utilizadas en esta variante son: número de etiquetas, demanda de cada etiqueta, capacidad de cada parrilla y número de parrillas (de acuerdo al estado del arte el número de parrillas varía de 3 a 6) (Yiu et. al, 2007), (Tuyttens & Vandaele, 2014), (Hsieh & You, 2014).

En la presente investigación se trabajó con el problema de impresión de etiquetas. Si analizamos el problema de manera cuidadosa y sistemática, la composición de cada parrilla a menudo se determina en función de la experiencia. Las diferentes combinaciones de etiquetas se asignan arbitrariamente a las parrillas hasta que se hayan asignado todas las etiquetas requeridas. La frecuencia de impresión de las etiquetas se determina luego para cumplir con el pedido del cliente. Claramente, este proceso requiere mucho esfuerzo humano y puede llevar a un gran desperdicio debido a la composición de parrillas deficientes (Yiu et. al, 2007).

Dado un número determinado de parrillas, el problema al que se enfrenta la imprenta consiste en configurar parrillas para imprimir etiquetas, minimizando el porcentaje de desperdicio.

Los elementos involucrados en este problema son los siguientes:

- Número de parrillas a utilizar.

- Composición de cada parrilla.
- Número de veces que se imprimirá cada parrilla.

Las restricciones principales a las que está sujeto se mencionan continuación:

- 1.- Cada etiqueta tiene una demanda que debe ser satisfecha.
- 2.- Cada etiqueta debe ir en una sola parrilla, sin embargo, pueden existir varias copias de una etiqueta en la parrilla.

## 1.3 Objetivos

### 1.3.1 General

Dado un número fijo de parrillas, el objetivo es minimizar el porcentaje de desperdicio satisfaciendo la demanda de cada etiqueta.

### 1.3.2 Específicos

- Realizar una revisión del estado del arte.
- Diseñar e implementar la metaheurística de Aceptación por Umbral.
- Diseñar e implementar la metaheurística de Búsqueda Tabú.
- Realizar experimentación con cada una de las instancias presentadas en el estado del arte.

## 1.4 Alcances y limitaciones

### Alcances

- Obtener soluciones satisfactorias de las 18 instancias disponibles en el estado del arte del problema de impresión de etiquetas.
- Validar los resultados obtenidos.

### Limitaciones

- Solo se utilizará el lenguaje c++ para el desarrollo de los algoritmos.
- Se trabajará solo con la variante impresión de etiquetas del problema de impresión.

## 1.5 Hipótesis

Es posible mejorar los resultados obtenidos en el estado del arte del problema de impresión de etiquetas utilizando los algoritmos de Aceptación por Umbral y de Búsqueda Tabú.

## 1.6 Justificación

Dada la dificultad práctica para resolver de forma exacta toda una serie de importantes problemas combinatorios, para los cuales es necesario ofrecer alguna solución debido a su interés práctico, comenzaron a aparecer algoritmos que proporcionan soluciones aproximadas, los cuales, aunque no garanticen encontrar el óptimo, al menos encuentran una solución “factible” en un tiempo de cálculo aceptable. Este tipo de algoritmos, basados en la intuición y estimaciones probabilísticas se denominan heurísticas. Unos de los factores que hacen interesante la utilización de algoritmos heurísticos para la resolución de un problema, son: Cuando no existe un método exacto de resolución o éste requiere mucho tiempo de cálculo o memoria o cuando no se necesita la solución óptima (Díaz et. al, 2000). La solución a problemas complejos, en particular los problemas NP-duro, implica un consumo de tiempo considerable durante su ejecución y de memoria de almacenamiento. Razón por la cual es de suma importancia desarrollar algoritmos que los resuelvan eficientemente.

En el problema de impresión de etiquetas se espera encontrar soluciones satisfactorias mediante los algoritmos de Aceptación por Umbral y Búsqueda tabú que generen soluciones de calidad en un tiempo razonable.

## 1.7 Organización de la tesis

En términos generales el trabajo de tesis está organizado de siguiente manera:

En el capítulo uno, se escribe la introducción, planteamiento del problema, objetivo general y objetivos específicos, justificación, alcances y limitaciones.

En el capítulo dos se describe el estado de arte del problema de impresión, se mencionan las diferentes metaheurísticas que se han propuesto para resolver la mayoría de las variantes del problema de impresión y los resultados que se han

obtenido. Para la variante de impresión de etiquetas sin y con restricción de repetición los mejores resultados se han obtenido fueron por medio del algoritmo basado en inmunidad.

Posteriormente en el capítulo tres se hace una descripción de las bases de las metaheurísticas de Aceptación por Umbral y Búsqueda Tabú utilizadas para la resolución del problema de impresión de etiquetas con restricción de repetición, así como también la descripción de los componentes que integran cada una de ellas.

En el capítulo cuatro se presenta de forma detallada la metodología que se utilizó para la solución del problema de impresión de etiquetas con restricción de repetición.

Continuando con el capítulo cinco se presenta la experimentación y los resultados obtenidos.

Finalmente, en el capítulo seis se describen las conclusiones y trabajos futuros.

## CAPÍTULO 2. MARCO TEÓRICO

### 2.1 Literatura del problema de impresión

En el año 1995 Teghem et. al. (Teghem et. al, 1995) publicaron por primera vez el problema de impresión, resolviendo la variante de problema de impresión de portadas. Propusieron las primeras cuatro instancias para este problema. Indican que los métodos exactos son incapaces de resolver dicho problema en un tiempo razonable, por lo que proponen una implementación de Recocido Simulado para resolver el problema. Ellos señalan la utilidad de las metaheurísticas como herramientas extremadamente eficientes para resolver problemas de optimización combinatoria. El Recocido Simulado se hibridó con técnicas de programación lineal (Método Simplex). Para desarrollar el algoritmo de Recocido Simulado y tener una mayor exploración de soluciones factibles, sugieren que se comience con una buena solución inicial, para generarla dividieron las portadas en tres tipos al momento de realizar la asignación de las portadas a las parrillas.

Tipo A.

Consiste en que aleatoriamente se asignan las  $m$  portadas diferentes en las  $n$  parrillas utilizadas, cada una con capacidad de  $t$  portadas. A los lugares de las parrillas con portadas asignadas se les denomina tipo A, a continuación, se muestra un ejemplo:

$$m = \{1,2,3,4\}.$$

$$n = 4.$$

$$t = 4.$$

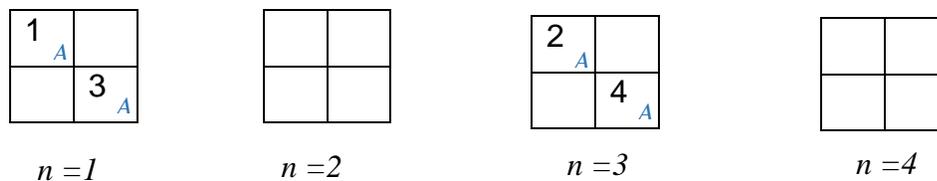


Figura 2. 1 representación tipo A

### Tipo B.

Identifica los espacios que aún no tienen asignada ninguna portada, a esos lugares se les denomina tipo B, aleatoriamente se asigna una portada a cada uno de ellos, esto es para que las parrillas que tienen lugares de tipo A queden configuradas totalmente.

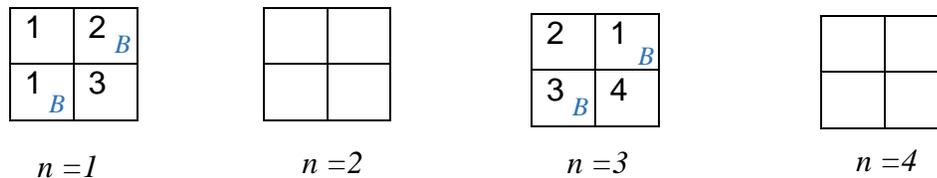


Figura 2. 2 Representación tipo B

### Tipo C.

Identifica las parrillas que están vacías, a estas parrillas se les denomina tipo C.

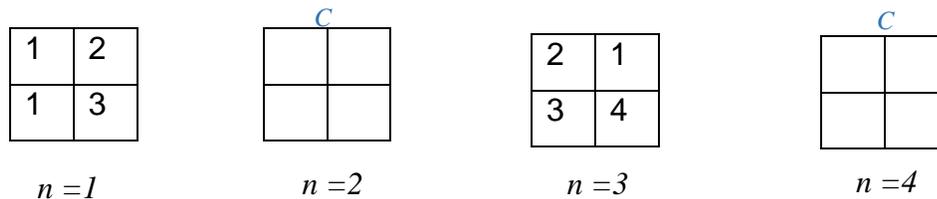


Figura 2. 3 Representación tipo C

Señalan que, al considerar estos tres tipos, cuando se generen las vecindades se tendrá una amplia búsqueda de soluciones factibles.

Considerando los resultados satisfactorios obtenidos por el algoritmo de Recocido Simulado, en el 2007 (Elaoud et. al, 2007), continuaron resolviendo el problema de impresión de portada, inspirados en la aplicación exitosa de algoritmos evolutivos para resolver problemas de optimización difíciles, se hizo una hibridación de Algoritmos Genéticos más programación lineal. Se utiliza el Algoritmo Genético (GA) para encontrar la mejor asignación de portadas a las parrillas y el número de parrillas a utilizar (costo de configuración de parrilla) y programación lineal para minimizar el

costo de impresión. De acuerdo a los resultados que obtuvieron, superaron los presentados por Teghem y colaboradores en las cuatro instancias y resolvieron dos instancias más, encontrando soluciones satisfactorias.

En ese mismo año, (Mohan et. al, 2007), comenzaron a abordar la variante de problema de impresión de etiquetas, consideran como un problema común de optimización que enfrenta una imprenta mientras diseña parrillas para imprimir material publicitario. Su objetivo era minimizar el desperdicio total, siempre y cuando se cubriera el total de la demanda de cada etiqueta; motivo por el cual abordan las siguientes restricciones: el límite inferior y límite superior del número de impresiones de una parrilla y etiquetas del mismo tamaño deben ir en una sola parrilla. Para lograr el objetivo proponen tres heurísticas a la medida, tanto obtener una buena solución inicial para posteriormente encontrar la mejor solución factible. A continuación, se describen brevemente las tres heurísticas (Mohan et. al, 2007):

1.- Heurística G-MIM (heurística codiciosa por número máximo de elementos por parrilla).

Esta heurística consiste en asignar el máximo número de etiquetas con mayor demanda en una parrilla, esto con el fin de minimizar el desperdicio y reducir el número de parrillas requeridas.

2.- Heurística de G-RBA (asignación basada en proporción codiciosa).

En esta heurística, se asignan las etiquetas en los espacios de las parrillas de acuerdo a la relación que existe entre la demanda de cada una. Para realizar la configuración se ordenan las etiquetas de menor a mayor demanda, en la parrilla actual se asigna la etiqueta con la demanda mínima, a partir de la primera asignación se establece un rango de aceptación para las siguientes asignaciones hasta que todas las etiquetas sean asignadas. Al aplicar esta heurística se esperaba obtener una buena solución inicial.

3.-Heurísticas basadas en vértices adyacentes

Este método de optimización basado en vértices adyacentes consiste en dos procedimientos. El proceso llamado descenso más empinado, comienza con una solución con el número mínimo de parrillas y busca minimizar el desperdicio. El

proceso dos, conocido como ascenso más plano, asume una solución en la que el desperdicio es pequeño, posteriormente el desperdicio puede ser un poco mayor para construir una solución, pero el número de parrillas utilizadas se reduce.

Al aplicar estas tres heurísticas al problema de impresión de etiquetas se resolvieron una instancia pequeña.

Continuando con la resolución a la variante de problema de impresión de etiquetas (Yiu et. al, 2007), consideraron que una buena configuración de la parrilla mejorará significativamente la eficiencia del proceso de impresión y, al mismo tiempo, reduce el desperdicio de recursos. Trabajaron con las restricciones que una etiqueta debe ir en una sola parrilla y se debe satisfacer la demanda de cada etiqueta, objetando que si una etiqueta era impresa en diferentes parrillas se incrementaba el costo de producción. Analizando el problema original derivan que se puede convertir en un problema de optimización de dos niveles. El nivel exterior optimiza la combinación de etiquetas, mientras que el nivel interior optimiza la asignación de las etiquetas asignadas en cada parrilla. Para encontrar una buena solución se utiliza el algoritmo de Recocido Simulado. Las iteraciones de Recocido Simulado se componen de pasos clave, a saber, la generación de vecindades en el espacio de la solución inicial a través de perturbaciones aleatorias, la elección de una distribución de probabilidad para regular la aceptación de soluciones y la sintonización de los parámetros: velocidad de enfriamiento, número de iteraciones para el enfriamiento y la temperatura inicial. Al aplicar el algoritmo de Recocido Simulado y utilizando el mayor número de parrillas posible, se reduce considerablemente el desperdicio. (Yiu et. al, 2007), Propusieron nuevas instancias de entre dieciocho y veintidós etiquetas, utilizando un número fijo de parrillas que va desde tres a cinco, con capacidad para catorce y quince etiquetas. El método propuesto es eficiente y los resultados calculados son consistentes; lo consideran como un medio simple pero poderoso para resolver el problema de impresión de etiquetas.

Para muchos problemas complejos de optimización de la vida real en diversas áreas, las metaheurísticas son técnicas que tienen un éxito generalizado en el

suministro de soluciones de alta calidad, casi óptimas para este tipo de problemas. El éxito de las metaheurísticas motivó a Tuyttens y Vandaele (Tuyttens & Vandaele, 2010), diseñar e implementar la heurística del Greedy Random Adaptative Search Procedure (GRASP) para el problema de impresión de portada en el 2010. El GRASP es una metaheurística de múltiples etapas, en la que cada iteración devuelve una solución viable al problema y consta de dos fases: una fase de construcción y una fase de búsqueda local. La mejor solución en todas las iteraciones se mantiene como resultado final. Consideraron nueve instancias del problema de impresión de portada; las primeras cuatro son instancias reales, la cinco y seis fueron presentadas cuando se resolvió esta variante a través de Algoritmos Genéticos y las instancias siete a nueve se construyeron con un mayor número de portadas para la experimentación del algoritmo propuesto. Los resultados obtenidos revelan que la metaheurística GRASP mejora las soluciones del Recocido Simulado y los Algoritmos Genéticos reportadas previamente en el estado del arte. Argumentaron que el GRASP podía seguir superando los resultados obtenidos.

En el 2010, Eikici et. al. (Ekici et. al, 2010), establecieron que el problema de impresión de portadas es un problema NP-duro. También lo denominaron problema de división del trabajo. Para resolver el problema de impresión de portadas proponen dos formulaciones de programación de enteros y dos heurísticas, la heurística de ejecución múltiple y la heurística de costo equilibrado. Estos algoritmos fueron aplicados a treinta y dos instancias reales, con un máximo de dos mil ochenta y seis portadas distintas. Las pruebas exhaustivas en instancias del mundo real y generadas aleatoriamente muestran que las heurísticas son rápidas y efectivas, encontrando soluciones cercanas a las óptimas.

Continuando con la búsqueda de soluciones factibles para el problema de impresión, en el año 2012, Romero y Alonso (Romero & Alonso, 2012), trabajaron con el problema de impresión de portadas, con el objetivo de minimizar el costo total de producción. Para resolver dicho problema, presentan una heurística a la medida,

que superó los resultados presentados por Tuyttens y Vandaele (Tuyttens & Vandaele, 2010). Se generan sesenta nuevas instancias y al aplicar el algoritmo se obtienen resultados satisfactorios para las mismas.

En 2014 el problema de impresión de etiquetas es abordado por Hsieh y You (Hsieh & You, 2014), para resolver el problema se aplica un algoritmo basado en inmunidad. La implementación se basa principalmente en permutaciones tomando en cuenta el número de etiquetas a imprimir y el número de parrillas utilizadas. Los resultados numéricos muestran que al aplicar dicho algoritmo se puede resolver el problema en un período de tiempo aceptable, obteniendo una solución satisfactoria, superando los resultados presentados por Yiu y colaboradores (Yiu et. al, 2007), al aplicar el algoritmo de Recocido Simulado. Propusieron doce nuevas instancias que van desde veintiséis a treinta y ocho etiquetas. Exponen que, si se usa un número de parrillas considerable, se disminuye el desperdicio total de etiquetas.

Nuevamente en ese mismo año 2014, Tuyttens y Vandaele (Tuyttens & Vandaele, 2014), abordaron las siguientes variantes del problema de impresión: el problema de impresión de etiquetas (con y sin restricción de repetición) y el problema de impresión de portada (sin costo de cliché). En ambas variantes, el problema consiste en la asignación de un número fijo de etiquetas en diferentes parrillas para cumplir con los objetivos de cada uno: minimizar el desperdicio de etiquetas y minimizar el costo total de producción. Ellos propusieron adecuar e implementar un GRASP para cada una de las variantes. Para el problema de impresión de etiquetas (en la que una etiqueta puede estar en una sola parrilla) se asegura que cada etiqueta se asigne a una sola parrilla desde la fase de construcción de la solución. En la fase de búsqueda local, solo se consideran los intercambios completos, lo que permite que no se viole la restricción. En la variante sin restricción de repetición de etiqueta, aumenta el número de formas posibles de asignar las etiquetas a las parrillas. Los resultados computacionales presentados muestran que la mejora al algoritmo GRASP produjo mejores resultados que los obtenidos por Yiu y colaboradores (Yiu et. al, 2007), en el caso de la variante de impresión de etiquetas que permite la

repetición de las mismas en varias parrillas. En el caso del problema de impresión de portadas sin costo cliché también mejoró los resultados presentados por Romero y Alonso (Romero, 2012). Concluyen que el GRASP es particularmente adecuado para este tipo de problemas.

En el 2018, (Alonso & Romero, 2018), propusieron una hibridación de una metaheurísticas más programación lineal para resolver el problema de impresión de portadas. Se efectúa una combinación de programación lineal con Recocido Simulado, así como una hibridación de Búsqueda Tabú y una heurística a la medida. Para desarrollar el algoritmo de Recocido Simulado se utilizaron cinco procedimientos principales, los cuales son: programación lineal (Método simplex), función de aleatoriedad, solución inicial, redimensionar y vecindad. La implementación de estos procedimientos en Recocido Simulado genera buenas soluciones al igual que Búsqueda Tabú y la heurística a la medida. En esta última hibridación, la heurística a la medida se utiliza para generar una solución inicial al problema, posteriormente a través de las instrucciones del algoritmo tabú, se van generando las vecindades para ir explorando el espacio de soluciones, cuando se encuentra una mejora, se acepta la solución; los elementos involucrados en esta mejora obtienen un estado el cual se denomina Tabú, por un cierto número de iteraciones al azar, el estado se va actualizando para evitar ciclos y quedarte atrapado en los mínimos locales. Al aplicar estas hibridaciones de metaheurísticas resultaron ser superiores en la mayoría de las instancias a los enfoques anteriores. Los resultados que obtuvieron son hasta la fecha los mejores presentados en el estado del arte.

## 2.2 Clasificación del problema de impresión

El problema de impresión consiste en configurar parrillas adecuadamente, de tal forma que se minimice ya sea el costo total de producción o el desperdicio generado. Es considerado como un problema de optimización combinatoria, perteneciente a la clase de complejidad NP-duro (Ekici et. al, 2010).

De acuerdo a la literatura el problema de impresión tiene cuatro variantes, estas se clasifican de acuerdo a su objetivo y sus diferentes restricciones. Las variantes son las siguientes: problema de impresión de portada (con y sin costo de cliché), problema de impresión de etiquetas (con y sin restricción de repetición de etiquetas).

### 2.2.1 Problema de impresión de portadas sin costo de cliché

El problema de impresión de portada sin costo de cliché (problema de impresión de portada para simplificar, dado que es la variante más común) tiene como objetivo minimizar el costo total de producción: dado por el costo total de las parrillas utilizadas (el número de parrillas utilizadas no es fijo) más el costo total de impresión. En este problema se tiene un conjunto de  $M$  portadas diferentes, cada portada  $i$  tiene una demanda  $d_i$ . Para satisfacer la demanda, la parrilla de impresión cuenta con una capacidad de impresión de  $t$  portadas por hoja de impresión. El costo por configurar una parrilla está dado por  $C_1$  y el costo por imprimir una hoja de papel está dado por  $C_2$ , normalmente  $C_1 \gg C_2$ . El problema de impresión de portadas consiste en satisfacer la demanda de producción minimizando la suma de los costos de impresión y de configuración de parrilla. Esta variante la han trabajado (Teghem et. al, 1995), (Elaoud et. al, 2007), (Ekici et. al, 2010), (Tuyttens & Vandaele, 2010), (Romero & Alonso, 2012), (Tuyttens & Vandaele, 2014), y (Alonso & Romero, 2018). El problema de impresión de portada puede plantearse como un problema de programación entera, la siguiente formulación la encontramos en (Alonso & Romero, 2018):

$$(\min) C_1 \sum_{j \in N} x_j + C_2 \sum_{j \in N} w_j$$

sujeto a

$$\sum_{j \in N} x_j a_{ij} \geq d_i \quad i \in M, \quad (1)$$

$$x_j \leq w_j d_m \quad j \in N, \quad (2)$$

$$x_j \geq 0 \text{ and integer } j \in N, \quad (3)$$

$$w_j \in \{0,1\}, \quad j \in N, \quad (4)$$

Dado

$M$  número de portadas.

$N$  número de parrillas.

$C_1$  costo de impresión.

$C_2$  costo para componer una parrilla.

$X_j$  número de impresiones.

$W_j$  variable de decisión.

$a_{ij}$  número de portada  $i$ -ésimo asignada a la  $j$ -ésima parrilla.

### 2.2.2 Problema de impresión de etiquetas con restricción de repetición

En el problema de impresión de etiquetas con restricción de repetición (mejor conocido como problema de impresión de etiqueta), su objetivo es minimizar el desperdicio total de etiquetas impresas. Dado un número determinado de  $n$  parrillas, el problema al que se enfrenta la imprenta consiste en configurar parrillas para imprimir un conjunto de  $m$  etiquetas diferentes, minimizando el desperdicio. Una restricción a tomar en cuenta es que cada etiqueta debe ir en una sola parrilla, sin embargo, pueden existir varias copias de una etiqueta en la parrilla. Cada etiqueta tiene una demanda  $d_i$  que debe ser satisfecha. Para satisfacer la demanda, cada una de las parrillas utilizadas tienen una capacidad para imprimir  $t$  etiquetas, la

asignación de etiquetas debe ser igual a la capacidad de la parrilla. Esta variante se abordó en (Yiu et. al, 2007) y (Hsieh & You 2014).

Para minimizar el desperdicio total de etiquetas dado un número fijo de parrillas, el problema de impresión de etiquetas se puede formular matemáticamente como:

$$\min \sum_{i=1}^m \sum_{j=1}^n x_{ij} k_j - \sum_{i=1}^m d_i$$

Sujeto a

$$\sum_{i=1}^m x_{ij} = t, \quad 1 \leq j \leq n, \quad (1)$$

$$x_{ij} \leq t y_{ij} \quad 1 \leq i \leq m, 1 \leq j \leq n, \quad (2)$$

$$\sum_{j=1}^n x_{ij} k_j \geq d_i, \quad 1 \leq i \leq m, \quad (3)$$

$$\sum_{j=1}^n y_{ij} = 1, \quad 1 \leq i \leq m, \quad (4)$$

$$y_{ij} \in \{0,1\}, \quad 0 \leq x_{ij} \leq t, \quad k_j \in N, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \quad (5)$$

Dado

- $n$  número de parrillas.
- $m$  número de etiquetas,  $m \geq n$ .
- $t$  capacidad de una parrilla.
- $d_i$  demanda de la etiqueta  $i$ .
- $k_j$  número de impresiones de la parrilla  $j$ .
- $x_{ij}$  número de etiqueta  $i$ -ésimo asignado a la  $j$ -ésima parrilla,  $1 \leq i \leq m, 1 \leq j \leq n$ .
- $y_{ij} = 1$  si la  $i$ -ésima etiqueta se asigna a la  $j$ -ésima parrilla, de lo contrario es 0.

A continuación, se muestra un pequeño ejemplo del modelo matemático, donde los valores de las variables son los siguientes:

$$m = \{1,2,3,4,5,6,7,8,9\}.$$

$$n = 3.$$

$$t = 6.$$

$$d_1 = 1000, d_2 = 2000, d_3 = 3000, d_4 = 4000, d_5 = 5000, d_6 = 6000, d_7 = 7000, d_8 = 8000, d_9 = 9000$$

Las siguientes expresiones describen las formulaciones matemáticas presentadas anteriormente como las restricciones de la función objetivo:

- (1): La sumatoria de los elementos de la matriz  $x_{ij}$  debe ser igual a la capacidad de la parrilla.
- (2): Existe una correspondencia entre la matriz  $x_{ij}$  y  $y_{ij}$ .
- (3): Se debe satisfacer la demanda de cada etiqueta.
- (4): Una etiqueta debe ir en una sola parrilla.

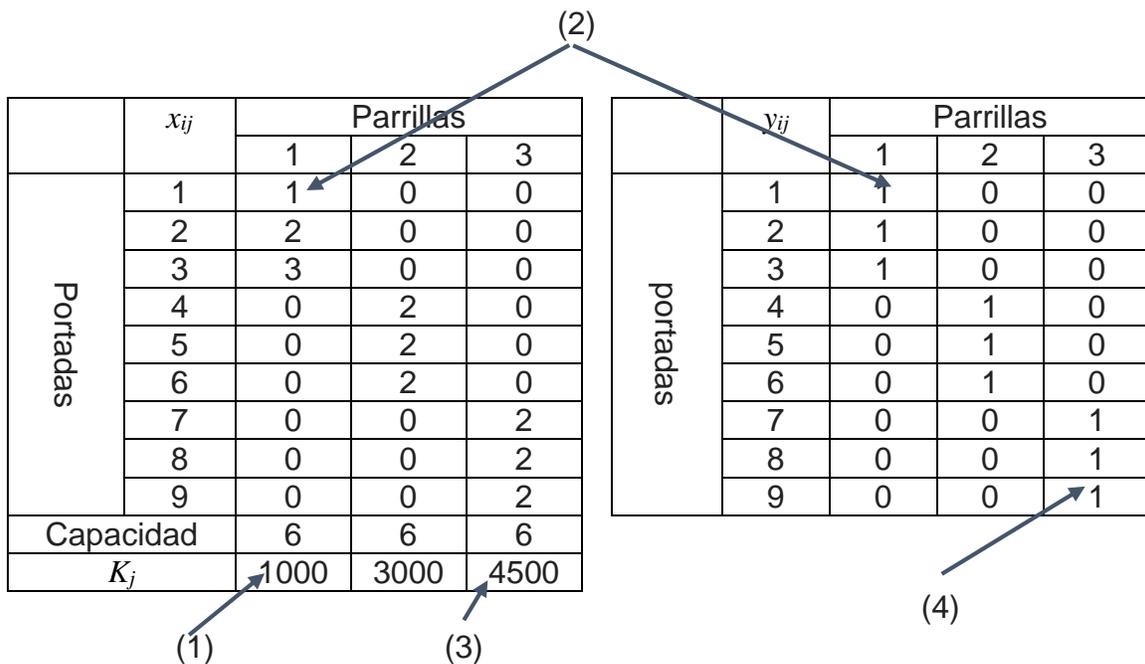


Figura 2. 4 Representación de restricciones

Una vez configuradas las parrillas se procede a calcular el desperdicio, para esto se multiplica en el número de asignaciones de la etiqueta  $x_{ij}$  por en número de impresiones de la parrilla  $K_j$ , al resultado obtenido se le resta la demanda de etiqueta  $d_i$  para obtener el desperdicio, en la tabla 2.1, se desglosan las operaciones correspondientes para calcular el desperdicio.

	$x_{ij}$			$K_j$	$x_{ij} K_j$	$d_i$	Desperdicio
	1	0	0	1000	1000	1000	0
	2	0	0	1000	2000	2000	0
	3	0	0	1000	3000	3000	0
	0	2	0	3000	6000	4000	2000
	0	2	0	3000	6000	5000	1000
	0	2	0	3000	6000	6000	0
	0	0	2	4500	9000	7000	2000
	0	0	2	4500	9000	8000	1000
	0	0	2	4500	9000	9000	0
Total						45000	6000

Tabla 2. 1 Desglose para calcular función objetivo

Para obtener el porcentaje de desperdicio se realiza la siguiente operación:

$$\text{Porcentaje} = \text{desperdicio} / d_i$$

$$\text{Porcentaje} = 6000 / 45000$$

$$\text{Porcentaje} = 0.13333 * 100$$

$$\text{Porcentaje} = 13.3 \%$$

### Espacio de soluciones

Para calcular el espacio de soluciones del problema de impresión de etiquetas, se necesitan los siguientes datos:

$n$  número de parrillas.

$m$  número de etiquetas.

$t$  capacidad de una parrilla.

a) Se tienen  $k$  divisiones de  $m$  de la siguiente manera:

$$n_1, n_2 \dots n_k$$

b) Cada una de las  $k$  divisiones representa un conjunto de combinaciones de parrilla, para sacar el total, deben multiplicarse todas:

$$\binom{m}{n_1} \binom{m - n_1}{n_2} \binom{m - n_1 - n_2}{n_3} \dots \binom{m - \sum_{i=0}^{k-1} n_i}{n_k}$$

c) Deducimos que, para calcular el espacio de soluciones, se utiliza la siguiente formula:

$$\prod_{i=1}^k \binom{m - \sum_{j=0}^{i-1} n_j}{n_i}$$

Donde

$$\sum_{i=1}^k n_i = m$$

En el siguiente ejemplo se desglosa la fórmula para obtener las combinaciones por cada configuración de parrillas. Tomando en cuenta la instancia LPP18 (ver tabla 5.1).

Donde los valores de las variables son los siguientes:

$$m = 18$$

$$n = 3$$

$$t = 14$$

1. Para la configuración de la primera parrilla se asignan 14 etiquetas, ya que de acuerdo a la capacidad de la parrilla es el máximo número de etiquetas que podemos asignar de un total de 18 etiquetas, realizando la siguiente combinación:

$$\binom{18}{14} = 3060 \text{ combinaciones.}$$

2. En la segunda parrilla se asignan 3 etiquetas tomando en cuenta las 4 etiquetas que hacen falta por asignar, haciendo la siguiente combinación:

$$\binom{4}{3} = 4 \text{ combinaciones.}$$

3. Finalmente, en la tercera parrilla se asigna 1 etiqueta, considerado que solo 1 etiqueta aún no ha sido asignada, en este caso la combinación queda de la siguiente manera:

$$\binom{1}{1} = 1 \text{ combinación.}$$

Para obtener el total de combinaciones para este ejemplo, se multiplica las combinaciones de cada una de las parrillas:  $(3060) (4) (1) = 12240$ .

A continuación, se presenta el total de combinaciones que se pueden obtener al calcular la formula descrita anteriormente, aplicándola a la instancia LPP18 (ver tabla 5.1).

<b>Etiquetas en parrilla 1</b>	<b>Etiquetas en parrilla 2</b>	<b>Etiquetas en parrilla 3</b>	<b>Total de combinaciones</b>
14	3	1	12240
14	2	2	18360
13	4	1	42840
13	3	2	85680
12	5	1	111384
12	4	2	278460
12	3	3	371280
11	6	1	222768
11	5	2	668304
11	4	3	1113840
10	7	1	350064
10	6	2	1225224
10	5	3	2450448
10	4	4	3063060
9	8	1	437580
9	7	2	1750320
9	6	3	4084080
9	5	4	6126120
8	8	2	1969110
8	7	3	5250960
8	6	4	9189180
8	5	5	11027016
7	7	4	10501920
7	6	5	14702688
6	6	6	17153136
			<b>92206062</b>

Tabla 2. 2 Espacio de soluciones de instancia LPP18

### 2.2.3 Problema de impresión de portada con costo de cliché

El problema de la impresión portada con costo de cliché consiste en diseñar la composición de las parrillas de impresión y determinar el número de veces que se imprimirá cada una, para satisfacer la demanda de diferentes portadas minimizando el costo total de producción; el costo total de producción proviene de tres costos: costo total de imprimir una hoja, costo total de configuración de parrilla y costo de fabricación de cliché. Esta variante solo la menciona (Romero & Alonso, 2012) sin trabajarla.

### 2.2.4 Problema de impresión de etiquetas sin restricción de repetición

El problema de impresión de etiquetas sin restricción de repetición es un caso general del problema de impresión de etiquetas, en donde cada etiqueta se puede asignar a diferentes parrillas, aumentando el número de formas posibles de asignar las etiquetas a las parrillas. El objetivo de este problema es minimizar el desperdicio de etiquetas impresas. Esta variante la abordaron (Mohan et. al, 2007), (Romero & Alonso 2012) y (Tuyttens & Vandaele, 2014), aunque de manera superficial.

## CAPITULO 3. METAHEURÍSTICAS UTILIZADAS

La necesidad de disponer de herramientas que permitan ofrecer soluciones (cercanas a las óptimas) a problemas reales, ha despertado el interés por los métodos heurísticos, estas herramientas han sido desarrolladas con el objetivo de ayudarnos en la resolución de muchos tipos de problemas, especialmente problemas de optimización (Siarry, 2016). Principalmente, se pueden distinguir dos tipos de problemas de optimización: problemas discretos y subconjunto de problemas con variables continuas. En la práctica, también se pueden encontrar problemas mixtos, que comprenden simultáneamente variables discretas y variables continuas. Siarry (2016), los define de la siguiente manera:

- Problemas de optimización discretos: para este tipo de problemas no se conoce un algoritmo polinomial exacto (es decir que el tiempo de cálculo sea proporcional a  $N^n$ , donde  $N$  es el número de parámetros desconocidos del problema y  $n$  es una constante entera). Este es el caso, en particular, de los problemas conocidos como NP-hard, para los que se ha conjeturado que no hay una constante  $n > 1$  para la cual el tiempo de solución esté limitado por un polinomio de grado  $n$ .
- Subconjunto de problemas de optimización con variables continuas: para este tipo de problemas no se conoce un algoritmo que permita encontrar un óptimo (la mejor solución posible) en un número finito de cálculos.

Se han hecho numerosos esfuerzos, para resolver estos dos tipos de problemas. En el campo de la optimización continua, hay un conjunto significativo de métodos tradicionales para obtener el óptimo, pero estas técnicas a menudo son ineficaces si la función objetivo no posee una propiedad estructural particular. Para el campo de optimización discreta, se han propuesto un gran número de heurísticas, que producen soluciones cercanas al óptimo, pero la mayoría de ellas fueron concebidas específicamente para un problema dado.

La creación y el diseño de las metaheurísticas marca una reconciliación de los dos dominios, pueden aplicarse a todo tipo de problemas discretos y también pueden adaptarse a problemas continuos.

Siarry (Siarry, 2016), presenta una clasificación general de los métodos de optimización mono-objetivo, de la siguiente manera:

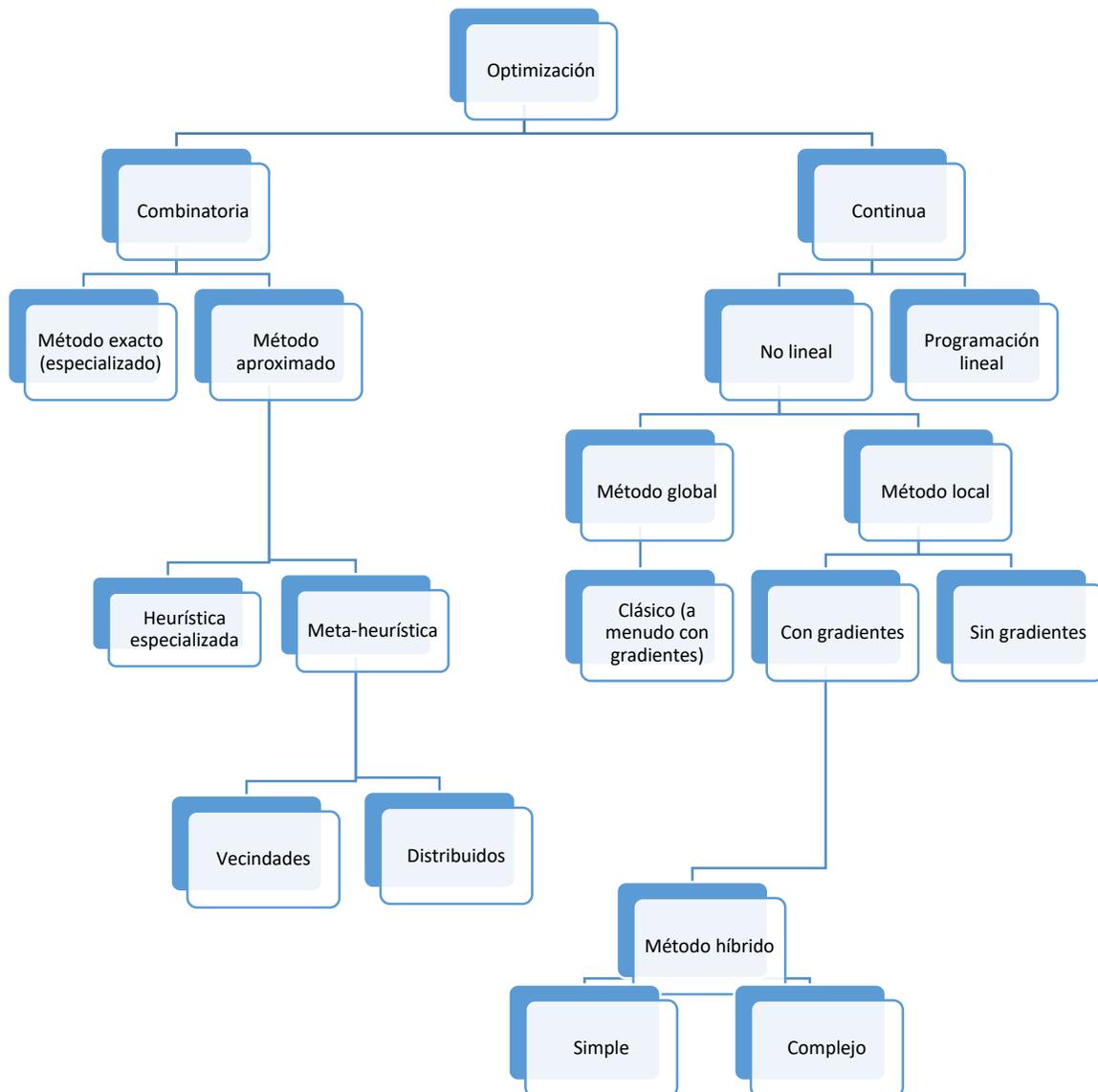


Figura 3. 1 Clasificación general de los métodos de optimización mono-objetivo

Las heurísticas se clasifican de acuerdo al modo en que buscan y construyen las soluciones, una posible clasificación es la siguiente según Díaz et. Al, (Díaz et. al, 2000):

- 1) Métodos constructivos: Consisten en ir añadiendo componentes individuales a la solución hasta que se obtiene una solución factible.
- 2) Métodos de descomposición: Se trata de dividir el problema en subproblemas más pequeños, siendo la salida de uno la entrada del siguiente, de forma que al resolverlos todos, se obtenga una solución para el problema global (divide y vencerás).
- 3) Métodos de reducción: tratan de identificar alguna característica que presumiblemente deba poseer la solución óptima y de ese modo simplificar el problema.
- 4) Manipulación del modelo: Estas heurísticas modifican la estructura del modelo con el fin de hacerlo más sencillo de resolver, deduciendo, a partir de su solución, la solución del problema original.
- 5) Métodos de búsqueda por entornos: En esta categoría es donde se encuentra la mayoría de las metaheurísticas. Estos métodos parten de una solución factible inicial y, mediante alteraciones de esa solución, van pasando de forma iterativa a otras factibles de su entorno, mientras no se cumpla un determinado criterio de parada, almacenando como óptima la mejor de las soluciones visitadas.

A pesar de que estos métodos heurísticos evalúan un número pequeño de alternativas del número total posible, han tenido bastante éxito en encontrar soluciones cercanas a las óptimas.

Una de las principales dificultades para poder resolver un problema de optimización, es saber elegir un método eficiente, capaz de producir una buena solución en un tiempo razonable. Generalmente se requiere del conocimiento y la experiencia del usuario para una buena elección.

A continuación, se describen las metaheurísticas que se utilizaron para la solución al problema de impresión de etiquetas, estas se encuentran dentro de la clasificación de métodos de búsqueda por entornos.

### 3.1 Aceptación por Umbral

La metaheurística de Aceptación por Umbral fue propuesta por Gunter Dueck y Tobías Scheuer en 1990 (Dueck, 1990). Es un algoritmo de propósito general para la solución de problemas de optimización combinatoria. La idea fundamental de esta clase de algoritmos es permitir movimientos hacia soluciones con valores de función objetivo más altos (movimientos ascendentes) para escapar de los mínimos locales. Las características principales que se esperan de este algoritmo, son las siguientes:

- Una buena aproximación al óptimo global.
- Que sea robusto al momento de ajustar los parámetros.
- Fácil de implementar para muchas instancias del problema.
- Búsqueda local similar al del Recocido Simulado.
- Permita movimientos ascendentes.
- Requiere estructura local en el espacio de búsqueda.
- Requiere una secuencia de umbral.

La implementación de dicho algoritmo implica la definición de vecindades (estructura local en el espacio de búsqueda, su valor objetivo debe estar cerca del valor anterior), una secuencia de umbral y la función objetivo (los límites inferiores teóricos de la función objetivo ayuda para evaluar el desempeño del algoritmo).

El procedimiento de Aceptación por Umbral es el siguiente: aleatoriamente de los vecinos  $N(s)$  de la solución actual  $s$  se extrae una solución  $s'$ . Se acepta  $s'$  si  $c(s') - c(s) < Th$ , donde  $Th$  es un parámetro de control llamado umbral. El umbral  $Th$  se reduce durante el proceso de búsqueda. Mientras el valor de  $Th$  sea alto, la búsqueda local realizada no está orientada a objetivos, logrando así una alta diversificación y una baja intensificación del proceso de búsqueda. Sin embargo, como el procedimiento de búsqueda evoluciona y  $Th$  se reduce, el equilibrio entre

diversificación e intensificación cambia. El algoritmo se comporta parecido como un algoritmo de búsqueda local descendente (Tarantilis, 2004).

A continuación, se describe de forma general el algoritmo de Aceptación por Umbral de acuerdo a Gilli, M. (Gilli, 2004):

```
1: Inicializar  $n_R$ ,  $n_S$  y  $Tr$ 
2: Generar solución actual  $s \in N(s)$ 
3: for  $r = 1$  to  $n_R$  do
4:   for  $i = 1$  to  $n_S$  do
5:     Generar  $s' \in N(s)$  (vecindad de solución actual)
6:     if  $f(s') < f(s) + Tr$  entonces
7:        $s = s'$ 
8:     termina if
9:   termina for
10: termina for
```

La estructura del algoritmo Aceptación por Umbral es más simple que la del algoritmo de Recocido Simulado. La metaheurística de Recocido Simulado traslada el proceso de recocido de metales para la resolver un problema de optimización: la función objetivo del problema, se minimiza, con la ayuda de un parámetro de control del algoritmo llamada temperatura.

Dentro del Recocido Simulado se trabaja con el algoritmo de Metrópolis, que permite describir el comportamiento de un sistema termodinámico en equilibrio a una cierta temperatura. Sobre la base de una configuración dada (por ejemplo, una colocación inicial de todos los componentes), el sistema está sujeto a una modificación elemental (por ejemplo, se puede reubicar un componente o intercambiar dos componentes). Si esta transformación hace que la función objetivo (o energía) del sistema disminuya, se acepta. Por otro lado, si causa un incremento  $\Delta E$  en la función objetivo, también puede aceptarse, pero con una probabilidad  $e^{-\frac{\Delta E}{T}}$ . Este proceso

se repite de manera iterativa, manteniendo la temperatura constante, hasta que se alcanza el equilibrio termodinámico, al final de un número “suficiente” de modificaciones. Luego, la temperatura disminuye antes de implementar una nueva serie de transformaciones: la regla por la cual la temperatura disminuye en etapas es a menudo empírica, al igual que el criterio para la terminación del programa (Siarry, 2016).

BoussaïD et. al (BoussaïD et. al, 2013), argumentan que el método de Aceptación por Umbral es una variante de Recocido Simulado. La principal diferencia entre Aceptación por Umbral y Recocido Simulado reside en el criterio de aceptación de las soluciones candidatas: por un lado, Recocido Simulado acepta una solución que causa el deterioro de la función objetivo  $f$  solo con cierta probabilidad; por otro lado, Aceptación por Umbral acepta esta solución si la degradación de  $f$  no supera un umbral  $T$  que disminuye progresivamente.

Se cree que para algunos problemas el algoritmo de Aceptación por Umbral supera al algoritmo de Recocido Simulado (Dueck, 1990).

### 3.2 Búsqueda Tabú

La metaheurística de Búsqueda Tabú (tabú search, TS, en inglés) fue propuesta por Fred Glover en 1986 (Glover, 1986). Búsqueda Tabú es un procedimiento metaheurístico utilizado para guiar un algoritmo heurístico de búsqueda local para explorar el espacio de soluciones más allá de la simple optimalidad local (Díaz et. al, 2000).

Búsqueda Tabú ha tenido mucho éxito en aplicaciones de prácticas de optimización. Lo que se necesita para la implementación del algoritmo Búsqueda Tabú son los siguientes elementos:

- Función objetivo: La función que nos permite identificar la calidad de la solución, generalmente es maximizar o minimizar una expresión lineal o no lineal.
- Vecindades: Las posibles soluciones cercanas a la solución inicial.

- Lista tabú: Lista de las posibles soluciones que no pueden ser elegidas en la actual iteración. Pueden ser las  $L$  soluciones anteriores o los  $L$  últimos movimientos.
- Conjunto de aspirantes o candidatos: Soluciones que tienen algún atributo que las hace elegibles aún si estuvieran en la lista tabú.
- Criterio de parada: Este criterio es necesario para evitar que el algoritmo entre en un proceso sin fin.

La implementación de una Búsqueda Tabú da lugar a un par de desafíos: primero, como en cualquier búsqueda iterativa, es necesario que el motor de búsqueda al momento de evaluar soluciones vecinas, sea efectivo; y, en segundo lugar, los conocimientos sobre el problema en cuestión deben transmitirse al procedimiento de búsqueda para que no quede atrapado en regiones poco prometedoras del espacio de la solución. Por el contrario, debe ser guiado inteligentemente en el espacio de solución. Glover propuso una serie de técnicas de aprendizaje que pueden integrarse en una búsqueda local. Se trata de construir una historia de la búsqueda iterativa o, de manera equivalente, equipar la búsqueda con la memoria (Siarry, 2016).

El objetivo de este método es explorar el espacio de búsqueda de las soluciones factibles por una secuencia de movimientos. Para escapar de un óptimo local, algunos movimientos son clasificados como prohibidos o tabú. La característica que distingue a la Búsqueda Tabú de las otras metaheurísticas de búsqueda por entornos es el uso de la memoria, la cual tiene una estructura basada en una lista tabú y unos mecanismos de selección del siguiente movimiento. La lista tabú es una lista en donde se registran aquellas soluciones o atributos de soluciones que no deben ser elegidas.

La longitud de la lista controla la memoria del proceso de búsqueda. Si la longitud de la lista es baja, la búsqueda se concentrará en áreas pequeñas del espacio de búsqueda. Por el contrario, una gran longitud obliga al proceso de búsqueda a explorar regiones más grandes, ya que prohíbe volver a visitar un número mayor de

soluciones. Esta longitud puede variar durante la búsqueda, lo que lleva a algoritmos más robustos.

El número de movimientos de tabú se puede modificar durante el proceso de búsqueda. Se pueden considerar varias metodologías para esta elección: el número se puede decidir al azar entre un límite inferior y uno superior, en cada iteración o después de un cierto número de iteraciones. Estos límites a menudo se pueden identificar fácilmente; también pueden aumentarse o disminuirse en función de las características observadas durante la búsqueda (Siarry, 2016).

La memoria usada en Búsqueda Tabú se puede clasificar en dos tipos: memoria de corto plazo y memoria de largo plazo. Cada tipo de memoria está acompañada de sus propias estrategias, se presenta la descripción de cada tipo de memoria de acuerdo a Díaz et. al (Díaz et. al, 2000):

### **Memoria de corto plazo**

La memoria de corto plazo más comúnmente usada, lleva la cuenta de los atributos de solución que han sido cambiados en el pasado reciente, es llamada memoria basada en hechos recientes. Para explotar esta memoria, los atributos seleccionados que se presentan en soluciones recientemente visitadas son designados como tabú activos, y las soluciones que contienen estos elementos son las que se convierten en tabú. Esto evita que algunas soluciones del pasado reciente pertenezcan a  $N^*(x)$  y por lo tanto de que sean revisitadas.

Los elementos que componen este tipo de memoria son los siguientes:

- Manejo de memoria basada en la recencia: el proceso se maneja creando una o más lista tabú, las cuales registran los atributos tabú activo, explícita e implícitamente identifican su estatus actual. Se denomina tenencia tabú a la duración que un atributo permanece tabú activo.
- Niveles de aspiración: el criterio de aspiración introduce un elemento importante de flexibilidad en la Búsqueda Tabú. El estatus tabú de una solución puede ser ignorado si ciertas condiciones se cumplen, en la forma de niveles de aspiración.

- Estrategias para la lista de candidato: Las estrategias para la lista de candidatos son usadas para restringir el número de soluciones examinadas en una iteración dada. Para los casos en los que  $N^*(x)$  es grande o la evaluación de sus elementos es costosa.

### **Memoria de largo plazo**

La metaheurística Búsqueda Tabú se vuelve más potente cuando se considera la memoria de largo plazo. Los componentes principales de este tipo de memoria son las estrategias de intensificación y las estrategias de diversificación.

- Las estrategias de intensificación están basadas en la modificación de reglas de elección de tal manera que favorezcan combinaciones de movimiento y características de solución que históricamente hayan sido buenas.
- Las estrategias de diversificación, están diseñadas para conducir la búsqueda hacia nuevas regiones. Con frecuencia están basadas en modificar las reglas de elección para llevar a la solución atributos que no hayan sido usados frecuentemente.

A continuación, se presenta el algoritmo de TS de acuerdo a BoussaïD et. al. (BoussaïD, 2013):

- 1: *Elegir al azar, una solución inicial en el espacio de búsqueda*
- 2: *Lista tabú  $\leftarrow \emptyset$*
- 3: *Mientras el criterio de parada no se cumple*
- 4: *selecciona la mejor solución  $s' \in N(s) \setminus \text{lista tabú}$*
- 5:  *$s \leftarrow s'$*
- 6: *actualizar lista tabú*
- 7: *Fin*
- 8: *Devolver la mejor solución encontrada*

Se puede observar que Búsqueda Tabú, se basa en la creencia de que la elección de una mala estrategia sistemática de búsqueda es más fructífera que la elección de una buena de tipo aleatorio. Hace uso de estructuras especiales de memoria y

de estrategias de búsqueda dinámica diseñadas para guiar el proceso de optimización de un modo eficiente (Díaz et. al, 2000).

La metaheurística Búsqueda Tabú ha tenido un destacado éxito para resolver problemas de optimización combinatoria, lo que ha provocado un crecimiento acelerado de Búsqueda Tabú en años recientes (Batista & Glover, 2006).

## CAPITULO 4. METODOLOGÍA

En el presente capítulo, se presenta la manera en que se resuelve el problema de impresión de etiquetas, el desarrollo se realiza con la implementación de los algoritmos de Aceptación por Umbral y Búsqueda Tabú.

Para comenzar a desarrollar la solución del problema de impresión, se genera una solución inicial factible, una vez generada, se busca optimizar la solución a través de las metaheurísticas propuestas.

A continuación, se esboza el esquema del trabajo realizado:

- *Diseño de un algoritmo para encontrar una solución inicial*
- *Diseño del algoritmo de Aceptación por Umbral, para el problema de impresión de etiquetas.*
- *Sintonización de parámetros del algoritmo de Aceptación por Umbral*
- *Diseño del algoritmo de doble vecindad, para el problema de impresión de etiquetas.*
- *Diseño del algoritmo de Búsqueda Tabú, para el problema de impresión de etiquetas.*
- *Sintonización de parámetros de Búsqueda Tabú*
- *Experimentación*

### 4.1 Solución inicial factible

Para generar la solución inicial factible, se utiliza una heurística ad hoc, la cual realiza la asignación de las  $m$  etiquetas a cada una de las  $n$  parrillas, tomando en cuenta las restricciones del problema de impresión de etiquetas, las cuales son:

- Una etiqueta debe ir en una sola parrilla.
- La sumatoria de las etiquetas asignadas en la parrilla debe ser igual a la capacidad  $t$  de la parrilla.
- Se debe satisfacer la demanda  $d_i$  de cada etiqueta.

Se presenta el pseudocódigo del algoritmo para generar la solución inicial:

1.- Para  $e = 0$  hasta  $m$  hacer

    hacer

$ban = false;$

$C1 = rand() \% n;$

        If(  $|C1| < t$  )

$C1 = C1 \cup e$

$ban = true;$

    mientras  $ban == false;$

2.- Para  $i = 0$  hasta  $n$  hacer

$P_i = \emptyset$

$\forall e \in C_i$  hacer

$P_i \leftarrow P_i + e;$

$\gamma_e \leftarrow 1;$

3.- function  $\Gamma(C_1, C_2, \dots, C_n)$

    Para  $i = 1$  hasta  $n$  hacer

        mientras  $|P_i| < t$  hacer

            inicio

$e \leftarrow \text{Max ind } d_e/\gamma_e \text{ para } e \in P_i$

$P_i \leftarrow P_i + e;$

$\gamma_e \leftarrow \gamma_e + 1;$

            fin

4.- function  $\Delta ()$

$tot \leftarrow \sum_{e \in M} d_e$

$imp \leftarrow 0$

    para  $i = 1$  hasta  $n$  hacer

$imp = imp + \text{Max } d_e/\gamma_e \text{ para } e \in P_i$

$imp \leftarrow imp \times t;$

$dif \leftarrow imp - tot;$

    por  $\leftarrow \frac{dif}{tot} \times 100;$

    return por;

El paso número uno del algoritmo es asignar aleatoriamente las  $m$  etiquetas a cada una de las  $n$  parrillas. Para cada etiqueta se genera un número aleatorio entre 0 y  $n-1$ , que representará la parrilla a la que se asignará la etiqueta, antes de asignar la etiqueta a la parrilla se verifica que el número de etiquetas asignados a una cada una de las parrillas sea menor a la capacidad  $t$ , si es menor se asigna la etiqueta a la parrilla, en caso contrario se genera otro número aleatorio.

En el paso número dos se agregan las  $m$  etiquetas a las parrillas, para llevar a cabo este paso, se hace un ciclo que va desde la primera parrilla hasta las  $n$  parrillas, se van agregando las etiquetas que corresponden a cada una de las parrillas. También se inicializa el contador de copias de cada etiqueta a 1, dado que hay una sola copia en la parrilla por cada etiqueta.

En el paso tres se utiliza la función  $\Gamma$ . Esta función se encarga de terminar de configurar la parrilla si aún quedan espacios por asignar. Dentro de la función se realiza un ciclo que va desde la parrilla uno hasta las  $n$  parrillas y mientras las etiquetas asignadas que hay en cada parrilla sea menor a la capacidad de la misma parrilla, se busca la etiqueta que necesite el mayor número de impresiones para cubrir su demanda, una vez identificada, se agrega una copia del elemento y se repite este procedimiento hasta que las parrillas tengan  $t$  copias de etiquetas cada una.

Una vez configuradas las parrillas, se procede a calcular el porcentaje de desperdicio, para esto se crea la función  $\Delta$ , dentro de la función se obtiene el total de la demanda de las etiquetas y el total de impresiones que se requieren con una solución  $S$  dada. Se multiplica el total de impresiones por la capacidad  $t$ , posteriormente al resultado de la multiplicación se le resta la demanda total, obteniendo el desperdicio. Para calcular el porcentaje de desperdicio se divide el desperdicio entre la demanda total y se multiplica por cien, la función retorna el porcentaje.

En el siguiente ejemplo, se hace una descripción detallada del procedimiento para obtener una solución inicial, tenemos 18 etiquetas con su respectiva demanda, 3 parrillas con una capacidad de 14 copias de etiquetas por parrillas:

$$m = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18\}$$

$$d_1 = 2200, d_2 = 200, d_3 = 500, d_4 = 100, d_5 = 250, d_6 = 550, d_7 = 550, d_8 = 550, d_9 = 2500, d_{10} = 2450, d_{11} = 350, d_{12} = 1150, d_{13} = 3850, d_{14} = 1400, d_{15} = 2700, d_{16} = 1400, d_{17} = 3050, d_{18} = 5550.$$

$$n = 3.$$

$$t = 14.$$

$\gamma_i$  = número de asignaciones

1. Aleatoriamente se asignan las  $m$  etiquetas a cada una de las  $n$  parrillas.

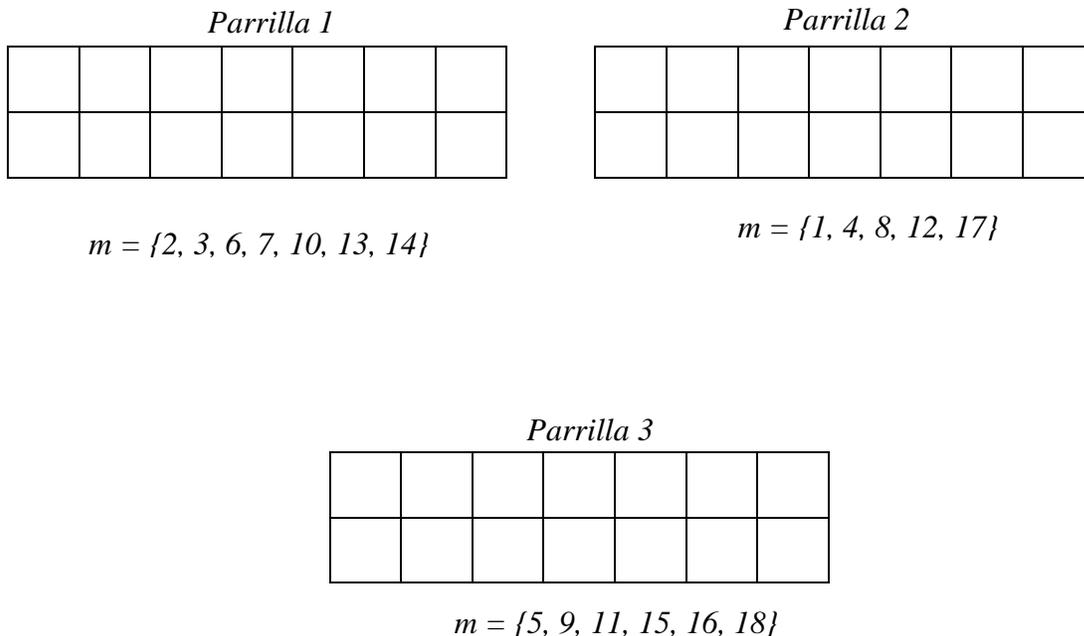


Figura 4. 1 Representación de asignación aleatoria de etiquetas a parrillas

2. De cada etiqueta asignada, se pondrá un elemento a la parrilla.

<i>Parrilla 1</i>							<i>Parrilla 2</i>						
2	3	6	7	10	13	14	1	4	8	12	17		

<i>Parrilla 3</i>						
5	9	11	15	16	18	

Figura 4. 2 Representación cuando se agregan las etiquetas a las parrillas

3. Si aún quedan espacios por asignar en la parrilla, se identifica a la etiqueta que requiera más impresiones para satisfacer su demanda, como se muestra en el inciso a y se agrega un elemento a la parrilla, tal como se muestra en el inciso b.

a)

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	1	1	1	1	1	1	1
$d_i / \gamma_i$	3850	2450	1400	550	550	500	200

*Parrilla 1*

b)

2	3	6	7	10	13	14
13						

Figura 4. 3 Identificación de la etiqueta con mayor demanda

3.1. Cuando la ya se agregó la copia de la etiqueta a la parrilla, se realiza una división para conocer el número de veces que se tiene que imprimir esa parrilla para

satisfacer la demanda de las etiquetas; se divide el total de la demanda de la etiqueta entre el número de asignaciones que tiene esa etiqueta en la parrilla.

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	2	1	1	1	1	1	1
$d_i / \gamma_i$	1925	2450	1400	550	550	500	200

Figura 4. 4 Resultado de la división entre la demanda de cada etiqueta y el número de asignaciones en la parrilla

Si aún quedan espacios por asignar en la parrilla, nuevamente se identifica a la etiqueta que requiera más impresiones para satisfacer su demanda, como se muestra en el inciso a y se agrega un elemento a la parrilla, como muestra en el inciso b. Cuando la ya se agregó la copia de la etiqueta a la parrilla, se realiza la división para conocer el número de veces que se tiene que imprimir esa parrilla para satisfacer la demanda de las etiquetas asignadas; se divide el total de la demanda de la etiqueta entre el número de asignaciones que tiene esa etiqueta en la parrilla, tal como se presenta en el inciso c.

a)

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	2	1	1	1	1	1	1
$d_i / \gamma_i$	1925	2450	1400	550	550	500	200

Parrilla 1

b)

2	3	6	7	10	13	14
13	10					

c)

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	2	3	1	1	1	1	1
$d_i / \gamma_i$	1925	1225	1400	550	550	500	200

Figura 4. 5 Identificación de etiqueta con mayor demanda, asignación de etiqueta y resultado de división

Continuando con la configuración se repiten los pasos anteriores.

a)

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	2	1	1	1	1	1	1
$d_i/\gamma_i$	1925	2450	1400	550	550	500	200

*Parrilla 1*

b)

2	3	6	7	10	13	14
13	10	13				

c)

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	3	2	1	1	1	1	1
$d_i/\gamma_i$	1284	1225	1400	550	550	500	200

Figura 4. 6 Identificación de etiqueta con mayor demanda, asignación de etiqueta y resultado de división

Se continúa realizando la asignación de las etiquetas a la parrilla, hasta que la parrilla quede configurada totalmente, y se identifica cuantas veces se tiene que imprimir la parrilla, de tal manera que satisfaga la demanda de las etiquetas que fueron asignadas.

$m$	13	10	14	7	6	3	2
$d_i$	3850	2450	1400	550	550	500	200
$\gamma_i$	5	3	2	1	1	1	1
$d_i/\gamma_i$	770	817	700	550	550	500	200

*Parrilla 1*

2	3	6	7	10	13	14
13	10	13	14	13	10	13

Se imprime 817 veces la parrilla 1

Figura 4. 7 Identificación de número de veces que se imprime la parrilla uno

3. Se realiza la configuración de la parrilla dos.

$m$	17	1	12	8	4
$d_i$	3050	2200	1150	550	100
$\gamma_i$	6	4	2	1	1
$d_i/\gamma_i$	509	550	575	550	550

*Parrilla 2*

1	4	8	12	17	17	17
1	1	1	12	17	17	17

Se imprime 575 veces  
la parrilla 2

Figura 4. 8 Configuración total de la parrilla dos

4. Se realiza la configuración de la parrilla tres.

$m$	18	15	9	16	11	5
$d_i$	5550	2700	2500	1400	350	250
$\gamma_i$	5	3	2	2	1	1
$d_i/\gamma_i$	1110	900	1250	700	350	250

*Parrilla 3*

5	9	11	15	16	18	18
18	9	15	15	16	18	18

Se imprime 1250  
veces la parrilla 3

Figura 4. 9 Configuración total de parrilla tres

5. Una vez terminada la configuración de la parrilla, se procede a calcular el desperdicio y su respectivo porcentaje. Para calcular el desperdicio se multiplica el número de asignaciones que tiene la etiqueta en la parrilla por el número de veces que se tiene que imprimir la parrilla, al resultado se le resta la demanda para obtener el desperdicio.

<i>Etiquetas</i>	<i>Num_asignaciones en parrillas</i>			<i>Num_impresiones de parrilla</i>	<i>Total de impresiones</i>	<i>Demanda</i>	<i>Desperdicio</i>
1	0	4	0	575	2300	2200	100
2	1	0	0	817	817	200	617
3	1	0	0	817	817	500	317
4	0	1	0	575	575	100	475
5	0	0	1	1250	1250	250	1000
6	1	0	0	817	817	550	267
7	1	0	0	817	817	550	267
8	0	1	0	575	575	550	25
9	0	0	2	1250	2500	2500	0
10	3	0	0	817	2451	2450	1
11	0	0	1	1250	1250	350	900
12	0	2	0	575	1150	1150	0
13	5	0	0	817	4085	3850	235
14	2	0	0	817	1634	1400	234
15	0	0	3	1250	3750	2700	1050
16	0	0	2	1250	2500	1400	1100
17	0	6	0	575	3450	3050	400
18	0	0	5	1250	6250	5550	700
					36988	29300	7688

Tabla 4. 1 Cálculo de porcentaje de ejemplo de solución inicial

Para calcular el porcentaje de desperdicio, se divide el desperdicio total entre la demanda total, como se muestra en el ejemplo:

$$\text{Porcentaje} = \text{desperdicio} / \text{demanda}$$

$$\text{Porcentaje} = 7688 / 29300$$

$$\text{Porcentaje} = 0.26238 * 100$$

$$\text{Porcentaje} = \mathbf{26.33 \%}$$

Después que se genera la solución inicial factible se implementan los algoritmos de Aceptación por Umbral y Búsqueda Tabú.

## 4.2 Aceptación por Umbral

En la implementación del algoritmo de Aceptación por Umbral para el problema de impresión de etiquetas, se tomaron en cuenta las siguientes características propias de la metaheurística Aceptación por Umbral: el vecindario de una solución, el umbral inicial, el factor de reducción del umbral, y el criterio de paro (umbral final).

En el siguiente pseudocódigo se describe el algoritmo general:

```
s_act = genera solución inicial
c_act = costo solución inicial
while (U >= U_MIN)
    cont = 0;
    while (cont < MAX)
        s_new = genera_vecino_aleatorio
        c_new = z(s_new)
        c_dif = c_new - c_act
        if (c_dif <= U)
            s_act = s_new
            c_act = c_new
            if (c_best > c_new)
                s_best = s_new
                c_best = c_new
        cont ++
    U = U * alpha
```

Para el algoritmo de Aceptación por Umbral se diseñaron dos vecindades a) vecindad de intercambiar y b) vecindad de agregar. La vecindad de intercambiar consiste en seleccionar aleatoriamente dos etiquetas, una vez seleccionadas se verifica que estén asignadas en diferente parrilla para realizar el intercambio. Con la nueva configuración que se obtuvo del intercambio, se recalcula la solución con la función  $\Gamma$ . La vecindad de agregar consiste en seleccionar aleatoriamente una etiqueta  $e_1$  que pertenece a una parrilla  $p_1$ . Si  $p_1$  tiene exactamente una etiqueta, se vuelve a seleccionar otra  $e_1$  aleatoria hasta que  $p_1$  tenga más de una etiqueta. Posteriormente se selecciona aleatoriamente una parrilla  $p_2$  tal que  $|p_2| < t$  y  $p_1 \neq p_2$ , se procede a cambiar a  $e_1$  de  $p_1$  a  $p_2$ , una vez que se obtiene la nueva configuración, se recalcula la solución con la función  $\Gamma$ .

Se presenta un ejemplo para describir las dos vecindades implementadas. Tenemos  $m$  etiquetas,  $n$  parrillas y la configuración inicial, la cual está representado por un arreglo del tamaño de las  $m$  etiquetas, donde se está mostrando en que parrilla fue asignada cada etiqueta.

$$m = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18\}$$

$$n = 3.$$

<b>n</b>	1	0	0	1	2	0	0	1	2	0	2	1	0	0	2	2	1	2
<b>m</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabla 4. 2 Configuración inicial

El criterio de vecindad, consiste en seleccionar aleatoriamente una de las dos vecindades implementadas. Si la vecindad seleccionada es la vecindad de intercambiar, se realizan los siguientes pasos:

- 1- Aleatoriamente se seleccionan dos etiquetas para intercambiarse.

<b>n</b>	1	0	0	1	2	0	0	1	2	0	2	1	0	0	2	2	1	2
<b>m</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabla 4. 3 Seleccionando etiquetas a intercambiarse

- 2- Se verifica que la etiquetas estén asignadas a diferentes parrillas.

<b>n</b>	1	0	0	1	2	0	0	1	2	0	2	1	0	0	2	2	1	2
<b>m</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabla 4. 4 Verificación de etiquetas

- 3- Se realiza el intercambio, si las etiquetas están asignadas a diferentes parrillas.

<b>n</b>	1	0	0	1	2	0	2	1	2	0	2	1	0	0	0	2	1	2
<b>m</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabla 4. 5 Nueva configuración

- 4- En dado caso de que las etiquetas estuvieren asignadas a la misma parrilla, nuevamente se repiten los pasos uno, dos y tres, hasta que cumpla con la

condición de que las etiquetas a intercambiar deben estar en diferente parrilla.

- 5- Posteriormente con la nueva configuración que se obtuvo del intercambio, se recalcula la solución con las funciones  $\Gamma$  y  $\Delta$ , el resultado obtenido es el costo de la solución de la vecindad. En seguida se evalúa a través del criterio de aceptación, si cumple con dicho criterio de aceptación, la nueva solución se convierte en la solución actual.

Si la vecindad seleccionada es la vecindad de agregar, se realizan los siguientes pasos:

- 1- Aleatoriamente se selecciona una etiqueta y una parrilla.

<b>n</b>	0	1	2
----------	---	---	---

<b>n</b>	1	0	0	1	2	0	2	1	2	0	2	1	0	0	0	2	1	2
<b>m</b>	1	2	3	4	5	6	15	8	9	10	11	12	13	14	7	16	17	18

*Tabla 4. 6 Configuración inicial*

- 2- Si  $p_1$  tiene exactamente una etiqueta, se vuelve a seleccionar otra  $e_1$  aleatoria hasta que  $p_1$  tenga más de una etiqueta. Posteriormente se selecciona aleatoriamente una parrilla  $p_2$  tal que  $|p_2| < t$  y  $p_1 \neq p_2$ .

<b>n</b>	0	1	2
----------	---	---	---

<b>n</b>	1	0	0	1	2	0	2	1	2	0	2	1	0	0	0	2	1	2
<b>m</b>	1	2	3	4	5	6	15	8	9	10	11	12	13	14	7	16	17	18

*Tabla 4. 7 Verificando que parrillas sean diferentes*

- 1- Si cumple la condición, se procede a cambiar a  $e_1$  de  $p_1$  a  $p_2$ , se agrega la etiqueta a la nueva parrilla.

<b>n</b>	1	1	0	1	2	0	2	1	2	0	2	1	0	0	0	2	1	2
<b>m</b>	1	2	3	4	5	6	15	8	9	10	11	12	13	14	7	16	17	18

Tabla 4. 8 Nueva configuración

- 2- En caso de no cumplir con la condición, se genera nuevamente de manera aleatoria la etiqueta y la parrilla, hasta que cumpla con la condición, que la parrilla seleccionada sea diferente a la parrilla de la etiqueta seleccionada. Esto es para agregar la etiqueta a la parrilla.
- 3- Una vez que se obtiene la nueva configuración, se recalcula la solución con las funciones  $\Gamma$  y  $\Delta$ , el resultado obtenido es el costo de la solución de la vecindad.

El criterio de aceptación de una nueva solución consiste en calcular la diferencia de la nueva solución que se obtuvo de las vecindades menos la solución actual. Si la diferencia es menor al umbral, se acepta la solución, y se convierte en solución actual.

Para la reducción del umbral, el umbral inicial, umbral mínimo y el número de iteraciones del ciclo, por medio de la sintonización experimental de parámetros se determinaron sus valores.

En el siguiente diagrama se resume lo implementado con la metaheurística de Aceptación por Umbral:



Figura 4. 10 Diagrama general de algoritmo Aceptación por Umbral

### 4.3 Sintonización de parámetros del algoritmo de Aceptación por Umbral

El uso de técnicas de ajuste de parámetros juega un papel cada vez más importante en el diseño, evaluación y aplicación de algoritmos de alto rendimiento para problemas difíciles (Hoos, 2011).

Para realizar la sintonización de parámetros del algoritmo de Aceptación por Umbral se consideraron los siguientes parámetros a sintonizar: umbral mínimo ( $u_{min}$ ), alpha ( $\alpha$ ) y la longitud del ciclo interno.

En la sintonización del parámetro de  $u_{min}$ , se probaron los valores  $\{0.0001, 0.00001\}$ , considerando que no se genere pérdida de calidad en la solución.

Por medio de sintonización experimental de parámetros se determina el valor de  $\alpha$  en el rango de (0,1). Se probaron los siguientes valores de acuerdo a Díaz et. Al, (Díaz et. al, 2000)  $\alpha = \{0.85, 0.90, 0.95, 0.99, 0.995, 0.999\}$ .

Se efectuó la sintonización de ambos parámetros para determinar el valor con el que se realizaría la experimentación. Se utilizó una de las instancias presentadas por Hsieh y You (Hsieh, 2014), LPP38 (ver tabla 5.1), esta instancia es la más grande, consta de treinta y ocho etiquetas y seis parrillas con capacidad para diecinueve etiquetas. Para cada valor asignado a  $\alpha$  y  $u_{min}$  se ejecutaron treinta corridas, en la tabla 4.9 se muestran los resultados de la sintonización:

<b>u_min \ alpha</b>	<b>0.85</b>	<b>0.9</b>	<b>0.95</b>	<b>0.99</b>	<b>0.995</b>	<b>0.999</b>
<b>0.0001</b>	3.841	3.450	3.621	3.743	3.865	<b>3.156</b>
<b>0.00001</b>	3.962	3.743	3.840	3.962	3.767	3.572

Tabla 4. 9 Resultados de sintonización de  $\alpha$  y  $u_{min}$

Como se aprecia en la tabla 4.9, el mejor resultado se obtiene, cuando de  $\alpha = 0.999$  y  $u_{min} = 0.0001$ , por lo tanto, estos parámetros se escogieron para la experimentación con las demás instancias.

Para hacer la sintonización del ciclo interno del algoritmo de Aceptación por Umbral, se probaron los valores  $\{500, 1000, 1500, 2000\}$ .

Se efectuó la sintonización de este parámetro para determinar el valor con el que se realizaría la experimentación. Se utilizó la misma instancia LPP38. Para cada número de iteraciones se ejecutaron treinta corridas, en la tabla 4.10 se muestran los resultados de la sintonización:

<b>Num_iteraciones</b>	<b>500</b>	<b>1000</b>	<b>1500</b>	<b>2000</b>
<b>Valor mínimo</b>	4.474	3.156	3.571	3.914
<b>Valor promedio</b>	4.924	4.327	4.921	4.971
<b>Tiempo promedio</b>	114.448	253.626	268.563	341.24

*Tabla 4. 10 Resultados de sintonización de ciclo interno*

En la tabla 4.10 de resultados, se puede observar que el mejor resultado se obtiene cuando en el ciclo interno se hacen mil iteraciones, por lo tanto, este parámetro se escogió para la experimentación con las demás instancias.

#### 4.4 Búsqueda local con doble vecindad

El algoritmo implementado de búsqueda local con doble vecindad es determinístico. Una vez que se tiene el resultado final (configuración y costo), ya sea del algoritmo Aceptación por Umbral o Búsqueda Tabú, se diseñaron dos tipos de doble vecindad a) doble vecindad de intercambiar y b) doble vecindad de agregar. La doble vecindad de intercambiar consiste en dada una solución actual  $s_{act}$ , se aplica dos veces la vecindad de intercambiar, se exploran todos los cambios posibles. La doble vecindad de agregar consiste en aplicar dos veces la vecindad de agregar a una solución  $s_{act}$ , nuevamente se exploran todos los cambios posibles. A continuación, se muestra un pseudocódigo de dicho algoritmo.



tenencia tabú y el criterio de paro. En el siguiente pseudocódigo se describe el algoritmo general:

```
f_solución_inicial(x)  
x_actual = x * = solución inicial  
LCandi =  $\emptyset$ ; LTabu =  $\emptyset$ ;  
mientras  
    LCandi = soluciones candidatas de la vecindad  
    x = mejor solución de LCandi  
    si  $f(x) < f(x^*)$   
        x_actual = x * = x  
        actualizar lista tabú  
    si no  
        x_actual = mejor vecino no tabú  
        actualizar lista tabú  
    terminará condición  
hasta criterio de paro  
función_doble_vecindad(z)
```

Para el algoritmo de Búsqueda Tabú, se implementaron las mismas dos vecindades que las del algoritmo de Aceptación por Umbral, solo que estas vecindades se generaron de forma determinística.

Para constituir la lista tabú, se crea un arreglo del tamaño de las  $m$  etiquetas, en dicho arreglo se guardan los últimos movimientos que no pueden ser elegidos en la actual iteración.

Para determinar la tenencia de la lista tabú se genera un número aleatorio entre un número determinado como límite superior y un número determinado como límite

inferior, este número representa el número de iteraciones que los movimientos permanecerán en la lista tabú.

Para determinar el número de iteraciones del ciclo externo del algoritmo, a través de la sintonización experimental de parámetros se determinó su valor.

#### 4.6 Sintonización de parámetros del algoritmo de Búsqueda Tabú

Para llevar a cabo la sintonización de parámetros del algoritmo de Búsqueda Tabú, se consideraron la tenencia tabú y el número de iteraciones del ciclo externo.

Para determinar el valor de la tenencia tabú, se toma en cuenta dos números arbitrarios, que funcionan como límite inferior y límite superior, se genera un número aleatorio entre esos límites. Para el valor del límite inferior se consideraron los siguientes valores {6, 7, 8, 9, 10}. Para el valor del límite superior se consideraron los siguientes valores {17, 18, 19, 20, 21}.

Se efectuó la sintonización de ambos parámetros para determinar los valores con el que se realizaría la experimentación. Se utilizó la misma instancia LPP38. Para cada valor asignado al límite inferior y límite superior se ejecutaron treinta corridas, en la tabla 4.11 se muestran los resultados de la sintonización:

<b>Lim_inferior</b>	<b>Lim_superior</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>
<b>6</b>		3.547	3.62	3.474	3.718	4.109
<b>7</b>		3.376	3.62	3.767	3.474	3.913
<b>8</b>		3.913	4.158	3.059	3.962	3.767
<b>9</b>		4.06	3.523	3.523	3.962	3.254
<b>10</b>		4.06	3.523	3.547	3.865	4.035

Tabla 4. 11 Resultados de sintonización de los límites inferior y superior

De acuerdo a la tabla de los resultados de sintonización de los límites inferior y superior, el mejor resultado se obtiene cuando el valor del límite inferior es ocho y el valor del límite superior es diecinueve. Por lo tanto, estos parámetros se escogieron para la experimentación con las demás instancias.

Para hacer la sintonización del ciclo externo del algoritmo de Búsqueda Tabú, se determinaron los valores {50, 100, 150, 200}.

Se efectuó la sintonización de este parámetro para determinar el valor con el que se realizaría la experimentación. Se utilizó la misma instancia LPP38. Para cada número de iteraciones se ejecutaron treinta corridas, en la tabla 4.12 se muestran los resultados de la sintonización:

<b>Num_iteraciones</b>	<b>50</b>	<b>100</b>	<b>150</b>	<b>200</b>
<b>Valor mínimo</b>	3.572	3.059	3.254	3.889
<b>Valor promedio</b>	4.911	4.919	4.591	4.561
<b>Tiempo promedio</b>	7.93	7.93	9.54	9.334

*Tabla 4. 12 Resultados de sintonización de ciclo externo*

De acuerdo a la tabla de los resultados de sintonización del ciclo externo, el mejor resultado se obtiene cuando el valor del número de iteraciones del ciclo externo es de cien. Por lo tanto, este parámetro se eligió para la experimentación con las demás instancias presentadas en el estado del arte.

De esta manera se realizó la sintonización de estos tres parámetros del algoritmo de Búsqueda Tabú.

## CAPITULO 5. EXPERIMENTACIÓN Y RESULTADOS

A continuación, se presentan los resultados que se obtuvieron al implementar los algoritmos de Aceptación por Umbral y Búsqueda Tabú.

Los algoritmos fueron probados para cada una de las instancias presentadas en el estado del arte del problema impresión de etiquetas. En la tabla 5.1 se presenta los resultados del algoritmo de Aceptación por Umbral, se presenta el mejor resultado, el peor resultado, la calidad promedio, tiempo promedio y desviación estándar. Se ejecutaron cien corridas para cada una de las instancias, las cuales son: LPP18, LPP22, LPP26, LPP30, LPP34 y LPP38. Los valores de los parámetros que se utilizaron en el algoritmo de Aceptación por Umbral son:  $u_{min} = 0.0001$ ,  $\alpha = 0.999$ , y el número de iteraciones del ciclo interno es de 1000.

<b>Instancia</b>	<b>m</b>	<b>t</b>	<b>n</b>	<b>Mejor</b>	<b>Peor</b>	<b>Calidad promedio</b>	<b>Desviación estándar</b>	<b>Tiempo promedio</b>
<b>LPP18</b>	<b>18</b>	<b>14</b>	<b>3</b>	<b>6.218</b>	<b>7.747</b>	<b>6.341</b>	<b>0.270</b>	<b>67.33</b>
			<b>4</b>	<b>3.352</b>	<b>5.358</b>	<b>4.347</b>	<b>0.493</b>	<b>90.72</b>
			<b>5</b>	<b>1.392</b>	<b>2.874</b>	<b>2.034</b>	<b>0.494</b>	<b>104.03</b>
<b>LPP22</b>	<b>22</b>	<b>15</b>	<b>3</b>	<b>8.152</b>	<b>10.254</b>	<b>8.907</b>	<b>0.590</b>	<b>74.75</b>
			<b>4</b>	<b>3.713</b>	<b>5.445</b>	<b>4.932</b>	<b>0.320</b>	<b>99.81</b>
			<b>5</b>	<b>2.215</b>	<b>4.107</b>	<b>3.174</b>	<b>0.437</b>	<b>115.9</b>
<b>LPP26</b>	<b>26</b>	<b>16</b>	<b>4</b>	<b>6.063</b>	<b>9.418</b>	<b>7.809</b>	<b>0.638</b>	<b>115.46</b>
			<b>5</b>	<b>3.507</b>	<b>5.983</b>	<b>4.556</b>	<b>0.611</b>	<b>126.56</b>
			<b>6</b>	<b>1.990</b>	<b>4.386</b>	<b>3.054</b>	<b>0.510</b>	<b>164.77</b>
<b>LPP30</b>	<b>30</b>	<b>17</b>	<b>4</b>	<b>7.182</b>	<b>11.706</b>	<b>9.498</b>	<b>0.868</b>	<b>119.45</b>
			<b>5</b>	<b>3.947</b>	<b>7.292</b>	<b>5.353</b>	<b>0.721</b>	<b>154.58</b>
			<b>6</b>	<b>2.192</b>	<b>5.592</b>	<b>3.622</b>	<b>0.703</b>	<b>197.19</b>
<b>LPP34</b>	<b>34</b>	<b>18</b>	<b>4</b>	<b>8.841</b>	<b>12.944</b>	<b>10.699</b>	<b>1.117</b>	<b>134.02</b>
			<b>5</b>	<b>3.977</b>	<b>7.395</b>	<b>5.829</b>	<b>0.890</b>	<b>174.91</b>
			<b>6</b>	<b>2.767</b>	<b>6.212</b>	<b>4.081</b>	<b>0.750</b>	<b>208.5</b>
<b>LPP38</b>	<b>38</b>	<b>19</b>	<b>4</b>	<b>9.335</b>	<b>16.051</b>	<b>12.345</b>	<b>1.546</b>	<b>148.6</b>
			<b>5</b>	<b>5.575</b>	<b>9.360</b>	<b>6.969</b>	<b>0.952</b>	<b>206.41</b>
			<b>6</b>	<b>3.157</b>	<b>7.333</b>	<b>4.551</b>	<b>0.769</b>	<b>254.58</b>

Tabla 5. 1 Resultados del algoritmo Aceptación por Umbral

Una vez terminada la experimentación con las dieciocho instancias presentadas en el estado del arte, se generaron seis nuevas instancias aleatorias: LPP42 y LPP46, para probar el tiempo que toman los algoritmos implementados con instancias más grandes que las del estado del arte. Se utilizaron los mismos valores para los parámetros. También se ejecutaron cien corridas para cada instancia.

En la tabla 5.2 se presentan los resultados que se obtuvieron, de las seis instancias generadas para experimentación del algoritmo de Aceptación por Umbral:

<b>Instancia</b>	<b>m</b>	<b>t</b>	<b>n</b>	<b>Mejor</b>	<b>Peor</b>	<b>Calidad promedio</b>	<b>Desviación estándar</b>	<b>Tiempo promedio</b>
<b>LPP42</b>	<b>42</b>	<b>20</b>	<b>4</b>	<b>9.833</b>	<b>18.700</b>	<b>12.204</b>	<b>1.501</b>	<b>157.86</b>
			<b>5</b>	<b>5.039</b>	<b>12.636</b>	<b>7.592</b>	<b>1.664</b>	<b>214.71</b>
			<b>6</b>	<b>4.016</b>	<b>9.428</b>	<b>5.761</b>	<b>0.976</b>	<b>343.16</b>
<b>LPP46</b>	<b>46</b>	<b>21</b>	<b>4</b>	<b>9.768</b>	<b>19.474</b>	<b>12.611</b>	<b>1.715</b>	<b>234.62</b>
			<b>5</b>	<b>6.224</b>	<b>13.001</b>	<b>8.350</b>	<b>1.351</b>	<b>274.98</b>
			<b>6</b>	<b>4.091</b>	<b>8.696</b>	<b>5.679</b>	<b>0.996</b>	<b>372.51</b>

*Tabla 5. 2 Resultados del algoritmo Aceptación por Umbral en las seis instancias generadas*

Para llevar a cabo la experimentación para el algoritmo de Búsqueda Tabú, se utilizaron las mismas instancias que se usaron para la experimentación del algoritmo de Aceptación por Umbral, estas son las dieciocho instancias que se presentan en el estado del arte y las seis generadas para una experimentación más detallada del algoritmo de Búsqueda Tabú. Se ejecutaron cien corridas para cada una. Los valores de los parámetros que se utilizaron en el algoritmo de Búsqueda Tabú son: *limite\_inferior = 8*, *limite\_superior = 14*, y el número de iteraciones del ciclo externo es de *100*, estos valores se manejaron para ejecutar cada una de las instancias.

En la tabla 5.3 se presentan los resultados que se obtuvieron, aplicando el algoritmo de Búsqueda Tabú:

<b>Instancia</b>	<b>m</b>	<b>t</b>	<b>n</b>	<b>Mejor</b>	<b>Peor</b>	<b>Calidad promedio</b>	<b>Desviación estándar</b>	<b>Tiempo promedio</b>
<b>LPP18</b>	<b>18</b>	<b>14</b>	<b>3</b>	<b>6.218</b>	<b>11.092</b>	<b>7.597</b>	<b>1.185</b>	<b>1.28</b>
			<b>4</b>	<b>3.352</b>	<b>6.792</b>	<b>5.012</b>	<b>0.688</b>	<b>1.17</b>
			<b>5</b>	<b>1.392</b>	<b>3.925</b>	<b>2.431</b>	<b>0.533</b>	<b>1.87</b>
<b>LPP22</b>	<b>22</b>	<b>15</b>	<b>3</b>	<b>8.152</b>	<b>13.879</b>	<b>9.593</b>	<b>1.170</b>	<b>1.56</b>
			<b>4</b>	<b>4.054</b>	<b>6.445</b>	<b>5.194</b>	<b>0.544</b>	<b>1.44</b>
			<b>5</b>	<b>2.425</b>	<b>4.869</b>	<b>3.401</b>	<b>0.508</b>	<b>1.84</b>
<b>LPP26</b>	<b>26</b>	<b>16</b>	<b>4</b>	<b>6.196</b>	<b>9.897</b>	<b>8.162</b>	<b>0.741</b>	<b>1.81</b>
			<b>5</b>	<b>3.108</b>	<b>6.383</b>	<b>4.524</b>	<b>0.640</b>	<b>2.71</b>
			<b>6</b>	<b>2.096</b>	<b>4.226</b>	<b>3.175</b>	<b>0.485</b>	<b>2.76</b>
<b>LPP30</b>	<b>30</b>	<b>17</b>	<b>4</b>	<b>7.620</b>	<b>11.898</b>	<b>9.445</b>	<b>0.921</b>	<b>1.86</b>
			<b>5</b>	<b>3.919</b>	<b>8.224</b>	<b>5.334</b>	<b>0.726</b>	<b>2.53</b>
			<b>6</b>	<b>2.411</b>	<b>6.497</b>	<b>3.575</b>	<b>0.601</b>	<b>2.84</b>
<b>LPP34</b>	<b>34</b>	<b>18</b>	<b>4</b>	<b>7.868</b>	<b>14.364</b>	<b>10.721</b>	<b>1.356</b>	<b>2.66</b>
			<b>5</b>	<b>4.292</b>	<b>7.974</b>	<b>5.904</b>	<b>0.799</b>	<b>3.84</b>
			<b>6</b>	<b>2.819</b>	<b>5.397</b>	<b>4.097</b>	<b>0.564</b>	<b>6.19</b>
<b>LPP38</b>	<b>38</b>	<b>19</b>	<b>4</b>	<b>9.530</b>	<b>16.540</b>	<b>11.939</b>	<b>1.559</b>	<b>3.62</b>
			<b>5</b>	<b>5.135</b>	<b>10.825</b>	<b>7.177</b>	<b>1.014</b>	<b>6.88</b>
			<b>6</b>	<b>3.059</b>	<b>7.748</b>	<b>4.979</b>	<b>0.710</b>	<b>7.93</b>

Tabla 5. 3 Resultados del algoritmo Búsqueda Tabú

En la tabla 5.4 se presentan los resultados que se obtuvieron, de las seis instancias generadas para experimentación del algoritmo de Búsqueda Tabú:

<b>Instancia</b>	<b>m</b>	<b>t</b>	<b>n</b>	<b>Mejor</b>	<b>Peor</b>	<b>Calidad promedio</b>	<b>Desviación estándar</b>	<b>Tiempo promedio</b>
<b>LPP42</b>	<b>42</b>	<b>20</b>	<b>4</b>	<b>9.217</b>	<b>16.497</b>	<b>12.204</b>	<b>1.162</b>	<b>7.04</b>
			<b>5</b>	<b>5.656</b>	<b>11.473</b>	<b>7.592</b>	<b>1.536</b>	<b>9.07</b>
			<b>6</b>	<b>3.999</b>	<b>7.868</b>	<b>5.761</b>	<b>0.762</b>	<b>10.20</b>
<b>LPP46</b>	<b>46</b>	<b>21</b>	<b>4</b>	<b>10.269</b>	<b>18.727</b>	<b>12.611</b>	<b>1.992</b>	<b>9.87</b>
			<b>5</b>	<b>5.863</b>	<b>16.307</b>	<b>8.350</b>	<b>1.352</b>	<b>12.69</b>
			<b>6</b>	<b>3.648</b>	<b>9.637</b>	<b>5.679</b>	<b>0.977</b>	<b>16.32</b>

Tabla 5. 4 Resultados del algoritmo Búsqueda Tabú en las seis instancias generadas

Una vez que se terminó con la experimentación de los algoritmos, se obtuvieron los mejores resultados de los algoritmos de Aceptación por Umbral y Búsqueda Tabú, y se hace una comparación contra los mejores resultados reportados en el estado del arte. En la tabla 5.5 se presentan los mejores resultados.

<i>Instancia</i>	<i>m</i>	<i>t</i>	<i>n</i>	<i>Mejor</i>	<i>Tiempo</i>	<i>Aceptación por umbral</i>	<i>Tabú</i>
<b>LPP18</b>	<b>18</b>	<b>14</b>	<b>3</b>	<b>6.218</b>	<b>705.77</b>	<b>6.218</b>	<b>6.218</b>
			<b>4</b>	<b>3.352</b>	<b>693.84</b>	<b>3.352</b>	<b>3.352</b>
			<b>5</b>	<b>1.392</b>	<b>703.72</b>	<b>1.392</b>	<b>1.392</b>
<b>LPP22</b>	<b>22</b>	<b>15</b>	<b>3</b>	<b>8.152</b>	<b>948.61</b>	<b>8.152</b>	<b>8.152</b>
			<b>4</b>	<b>3.581</b>	<b>830.95</b>	<b>3.713</b>	<b>4.054</b>
			<b>5</b>	<b>1.979</b>	<b>859.79</b>	<b>2.215</b>	<b>2.425</b>
<b>LPP26</b>	<b>26</b>	<b>16</b>	<b>4</b>	<b>6.516</b>	<b>1071.01</b>	<b>6.063</b>	<b>6.196</b>
			<b>5</b>	<b>3.667</b>	<b>1033.92</b>	<b>3.507</b>	<b>3.108</b>
			<b>6</b>	<b>2.309</b>	<b>1031.01</b>	<b>1.990</b>	<b>2.096</b>
<b>LPP30</b>	<b>30</b>	<b>17</b>	<b>4</b>	<b>8.169</b>	<b>1306.76</b>	<b>7.182</b>	<b>7.620</b>
			<b>5</b>	<b>4.111</b>	<b>1235.70</b>	<b>3.947</b>	<b>3.919</b>
			<b>6</b>	<b>2.384</b>	<b>1229.33</b>	<b>2.192</b>	<b>2.411</b>
<b>LPP34</b>	<b>34</b>	<b>18</b>	<b>4</b>	<b>9.025</b>	<b>1612.53</b>	<b>8.841</b>	<b>7.868</b>
			<b>5</b>	<b>4.161</b>	<b>1448.63</b>	<b>3.977</b>	<b>4.292</b>
			<b>6</b>	<b>3.109</b>	<b>1397.81</b>	<b>2.767</b>	<b>2.819</b>
<b>LPP38</b>	<b>38</b>	<b>19</b>	<b>4</b>	<b>10.141</b>	<b>1899.32</b>	<b>9.335</b>	<b>9.530</b>
			<b>5</b>	<b>5.454</b>	<b>1700</b>	<b>5.575</b>	<b>5.135</b>
			<b>6</b>	<b>3.889</b>	<b>1591.07</b>	<b>3.157</b>	<b>3.059</b>

*Tabla 5. 5 Comparación de mejores resultados de los algoritmos de Aceptación por Umbral, Búsqueda Tabú y el mejor del estado de arte*

En tabla de resultados presentada, se observa que para instancia LPP18, tanto el algoritmo de Aceptación por Umbral como el algoritmo de Búsqueda Tabú, encontraron el resultado óptimo para las tres primeras instancias.

Para la instancia LPP22, los algoritmos de Aceptación por Umbral y Búsqueda Tabú encuentran el resultado óptimo para la instancia de veintidós etiquetas con tres parrillas.

En la instancia LPP26, los algoritmos lograron superar al mejor resultado presentado en el estado del arte para las tres instancias de veintiséis etiquetas, con cuatro, cinco y seis parrillas.

Para la instancia LPP30, el algoritmo de Aceptación por Umbral mejoró los resultados presentados en el estado del arte.

Nuevamente para instancia LPP34, el algoritmo de Aceptación por Umbral mejora los resultados presentados en el estado del arte de las tres instancias.

Para la última instancia LPP38, los dos algoritmos mejoran resultados presentados en el estado del arte, el algoritmo de Aceptación por Umbral, supera tanto a los resultados de literatura como a los de Búsqueda Tabú, en la instancia de treinta y ocho etiquetas y cuatro parrillas; sin embargo, el algoritmo de Búsqueda Tabú supera los mejores resultados presentados en el estado del arte y los de Aceptación por Umbral, en las instancias de treinta y ocho etiquetas, cinco y seis parrillas.

En resumen, de las 18 instancias presentadas en el estado del arte, se mejoraron 12 resultados, se igualaron 4 y se perdió en 2. Los tiempos utilizados por los algoritmos son bastante competitivos, mejorando la mayoría de las instancias.

Del total de instancias trabajadas (dieciocho de la literatura y seis generadas) Aceptación por Umbral ganó en nueve instancias, y Búsqueda Tabú obtuvo el mejor valor en otras nueve, como se observa en la siguiente tabla.

<i>Instancia</i>	<i>m</i>	<i>t</i>	<i>n</i>	<i>Mejor</i>	<i>Tiempo</i>	<i>Aceptación por umbral</i>	<i>Tabú</i>
<i>LPP18</i>	<i>18</i>	<i>14</i>	<i>3</i>	<i>6.218</i>	<i>705.77</i>	<i>6.218</i>	<i>6.218</i>
			<i>4</i>	<i>3.352</i>	<i>693.84</i>	<i>3.352</i>	<i>3.352</i>
			<i>5</i>	<i>1.392</i>	<i>703.72</i>	<i>1.392</i>	<i>1.392</i>
<i>LPP22</i>	<i>22</i>	<i>15</i>	<i>3</i>	<i>8.152</i>	<i>948.61</i>	<i>8.152</i>	<i>8.152</i>
			<i>4</i>	<i>3.581</i>	<i>830.95</i>	<i>3.713</i>	<i>4.054</i>
			<i>5</i>	<i>1.979</i>	<i>859.79</i>	<i>2.215</i>	<i>2.425</i>
<i>LPP26</i>	<i>26</i>	<i>16</i>	<i>4</i>	<i>6.516</i>	<i>1071.01</i>	<i>6.063</i>	<i>6.196</i>
			<i>5</i>	<i>3.667</i>	<i>1033.92</i>	<i>3.507</i>	<i>3.108</i>
			<i>6</i>	<i>2.309</i>	<i>1031.01</i>	<i>1.990</i>	<i>2.096</i>
<i>LPP30</i>	<i>30</i>	<i>17</i>	<i>4</i>	<i>8.169</i>	<i>1306.76</i>	<i>7.182</i>	<i>7.620</i>
			<i>5</i>	<i>4.111</i>	<i>1235.70</i>	<i>3.947</i>	<i>3.919</i>
			<i>6</i>	<i>2.384</i>	<i>1229.33</i>	<i>2.192</i>	<i>2.411</i>
<i>LPP34</i>	<i>34</i>	<i>18</i>	<i>4</i>	<i>9.025</i>	<i>1612.53</i>	<i>8.841</i>	<i>7.868</i>
			<i>5</i>	<i>4.161</i>	<i>1448.63</i>	<i>3.977</i>	<i>4.292</i>
			<i>6</i>	<i>3.109</i>	<i>1397.81</i>	<i>2.767</i>	<i>2.819</i>
<i>LPP38</i>	<i>38</i>	<i>19</i>	<i>4</i>	<i>10.141</i>	<i>1899.32</i>	<i>9.335</i>	<i>9.530</i>
			<i>5</i>	<i>5.454</i>	<i>1700</i>	<i>5.575</i>	<i>5.135</i>
			<i>6</i>	<i>3.889</i>	<i>1591.07</i>	<i>3.157</i>	<i>3.059</i>
<i>LPP42</i>	<i>42</i>	<i>20</i>	<i>4</i>	<i>---</i>	<i>---</i>	<i>9.833</i>	<i>9.217</i>
			<i>5</i>	<i>---</i>	<i>---</i>	<i>5.039</i>	<i>5.656</i>
			<i>6</i>	<i>---</i>	<i>---</i>	<i>4.016</i>	<i>3.999</i>
<i>LPP46</i>	<i>46</i>	<i>21</i>	<i>4</i>	<i>---</i>	<i>---</i>	<i>9.768</i>	<i>10.269</i>
			<i>5</i>	<i>---</i>	<i>---</i>	<i>6.224</i>	<i>5.863</i>
			<i>6</i>	<i>---</i>	<i>---</i>	<i>4.091</i>	<i>3.648</i>

Tabla 5. 6 Comparación de mejores resultados de los algoritmos de Aceptación por Umbral, Búsqueda Tabú y el mejor de la literatura , del total de instancias

## CAPITULO 6. CONCLUSIONES Y TRABAJOS FUTUROS

En el presente capítulo se presentan las conclusiones del trabajo de investigación y los trabajos futuros.

### 6.1 Conclusiones

Con los algoritmos de Aceptación por Umbral, Búsqueda Tabú y búsqueda local con doble vecindad se logró mejorar doce de dieciocho instancias de la literatura, se igualaron cuatro y no se mejoraron dos.

Del total de instancias trabajadas (dieciocho de la literatura y seis generadas) Aceptación por Umbral ganó en nueve instancias, y Búsqueda Tabú obtuvo el mejor valor en otras nueve.

Analizando los resultados obtenidos, se concluye que estos dos algoritmos son adecuados para resolver el problema de impresión de etiquetas, ya que las soluciones obtenidas son de muy buena calidad, y el tiempo para resolver este problema va de uno a siete minutos al ejecutar el algoritmo de Aceptación por Umbral y de uno a siete segundos al ejecutar el algoritmo de Búsqueda Tabú, en cambio los tiempos de ejecución presentados en el estado del arte, van desde once hasta treinta y un minutos.

### 6.2 Trabajos futuros

Los resultados obtenidos al implementar estos algoritmos de aceptación, por umbral, Búsqueda Tabú y búsqueda local con doble vecindad, son satisfactorios, pero no se logró mejorar totalmente las dieciocho instancias presentadas en el estado del arte. Se propone hacer una mejora a estos algoritmos, los trabajos futuros son los siguientes:

- ❖ La propuesta es hacer doble vecindad haciendo una combinación de vecinos, esto es: primero generar el vecino agregar, posteriormente generar la vecindad de intercambiar e inversamente generar la vecindad de intercambiar y posteriormente la vecindad de agregar.
- ❖ Al algoritmo de Aceptación por Umbral, experimentar con otro valor de umbral inicial.
- ❖ Utilizar otra metaheurística para resolver el problema.

## REFERENCIAS

- [1] Alonso-Pecina, F., & Romero, D. (2018). A Hybrid Simulated Annealing/Linear Programming Approach for the Cover Printing Problem. *Mathematical Problems in Engineering*, 2018. doi.org/10.1155/2018/6193649
- [2] Batista, B. M., & Glover, F. (2006). Introducción a la búsqueda Tabú. vol, 3, 1-36.
- [3] BoussaïD, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information sciences*, 237, 82-117.
- [4] Bratley, P., & Brassard, G. (1997). *Fundamentos de algoritmia*. Madrid: Pearson Prentice-Hall.
- [5] Díaz, A., Glover, F., Ghaziri, H. M., González, J. L., Laguna, M., Moscato, P., & Tseng, F. T. (2000). Optimización heurística y redes neuronales.
- [6] Dueck, G., & Scheuer, T. (1990). Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of computational physics*, 90(1), 161-175.
- [7] Ekici, A., Ergun, Ö., Keskinocak, P., & Lagoudakis, M. G. (2010). Optimal job splitting on a multi-slot machine with applications in the printing industry. *Naval Research Logistics (NRL)*, 57(3), 237-251. DOI 10.1002/nav.20396
- [8] Elaoud, S., Teghem, J., & Bouaziz, B. (2007). Genetic algorithms to solve the cover printing problem. *Computers & operations research*, 34(11), 3346-3361. doi:10.1016/j.cor.2006.02.001
- [9] Gilli, M. (2004, September). An introduction to optimization heuristics. In Seminar university of cyprus department of public and business administration.
- [10] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5), 533-549.
- [11] Hoos, H. H. (2011). Automated algorithm configuration and parameter tuning. In *Autonomous search* (pp. 37-71). Springer, Berlin, Heidelberg.
- [12] Hsieh, Y. C., & You, P. S. (2014). An immune evolutionary approach for the label printing problem. *International Journal of Computational Intelligence Systems*, 7(3), 515-523. ISSN: 1875-6891.

- [13] Mohan, S. R., Neogy, S. K., Seth, A., Garg, N. K., & Mittal, S. (2007). An optimization model to determine master designs and runs for advertisement printing. *Journal of Mathematical Modelling and Algorithms*, 6(2), 259-271. DOI 10.1007/s10852-006-9039-0
- [14] Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152(1), 148-158.
- [15] Teghem, J., Pirlot, M., & Antoniadis, C. (1995). Embedding of linear programming in a simulated annealing algorithm for solving a mixed integer production planning problem. *Journal of Computational and Applied Mathematics*, 64(1-2), 91-102.
- [16] Tuytens, D., & Vandaele, A. (2010). Using a greedy random adaptive search procedure to solve the cover printing problem. *Computers & Operations Research*, 37(4), 640-648. doi:10.1016/j.cor.2009.06.006
- [17] Tuytens, D., & Vandaele, A. (2014). Towards an efficient resolution of printing problems. *Discrete Optimization*, 14, 126-146. doi.org/10.1016/j.disopt.2014.09.001
- [18] Romero, D., & Alonso-Pecina, F. (2012). Ad hoc heuristic for the cover printing problem. *Discrete Optimization*, 9(1), 17-28. doi:10.1016/j.disopt.2011.10.002
- [19] Siarry, P. (Ed.). (2016). *Metaheuristics (Vol. 23)*. Switzerland: Springer.
- [20] Yiu, K. F. C., Mak, K. L., & Lau, H. Y. (2007). A heuristic for the label printing problem. *Computers & operations research*, 34(9), 2576-2588. doi:10.1016/j.cor.2005.09.021



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



Cuernavaca, Morelos a 02 de Septiembre del 2019.

**DRA. LAURA PATRICIA CEBALLOS GILES**  
**DIRECTORA DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado Maestría en Optimización y Cómputo Aplicado, de la estudiante Rocio Diego Celis, con matrícula 10010396, con el título **DOS ENFOQUES METAHEURÍSTICOS PARA RESOLVER EL PROBLEMA DE IMPRESIÓN DE ETIQUETAS** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
*Por una humanidad culta*  
*Una universidad de excelencia*

**Dr. FEDERICO ALONSO PECINA**  
Profesor- investigador  
Facultad de Contaduría, Administración e Informática



Cuernavaca, Morelos a 04 de septiembre del 2019.

**DRA. LAURA PATRICIA CEBALLOS GILES**  
**DIRECTORA DE LA FCAEI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado Maestría en Optimización y Cómputo Aplicado, de la estudiante Rocio Diego Celis, con matricula 10010396, con el título **DOS ENFOQUES METAHEURÍSTICOS PARA RESOLVER EL PROBLEMA DE IMPRESIÓN DE ETIQUETAS**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
*Por una humanidad culta*  
*Una universidad de excelencia*



**Dr. LUIS MANUEL GAGGERO SAGER**  
Profesor-investigador  
Centro de Investigación en Ingeniería y Ciencias Aplicadas



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



Cuernavaca, Morelos a 02 de septiembre del 2019.

**DRA. LAURA PATRICIA CEBALLOS GILES**  
**DIRECTORA DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado Maestría en Optimización y Cómputo Aplicado, de la estudiante Rocio Diego Celis, con matricula 10010396, con el título **DOS ENFOQUES METAHEURÍSTICOS PARA RESOLVER EL PROBLEMA DE IMPRESIÓN DE ETIQUETAS**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
*Por una humanidad culta*  
*Una universidad de excelencia*

**Dr. JOSÉ ALBERTO HERNÁNDEZ AGUILAR**  
Profesor- investigador  
Facultad de Contaduría, Administración e Informática



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



## FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a 24 de Septiembre del 2019.

**DRA. LAURA PATRICIA CEBALLOS GILES**  
**DIRECTORA DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado Maestría en Optimización y Cómputo Aplicado, de la estudiante Rocio Diego Celis, con matrícula 10010396, con el título **DOS ENFOQUES METAHEURÍSTICOS PARA RESOLVER EL PROBLEMA DE IMPRESIÓN DE ETIQUETAS** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
*Por una humanidad culta*  
*Una universidad de excelencia*

**Dr. MARTÍN HERIBERTO CRUZ ROSALES**  
**Profesor- investigador**  
**Facultad de Contaduría, Administración e Informática**



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a 26 de agosto del 2019.

**DRA. LAURA PATRICIA CEBALLOS GILES**  
**DIRECTORA DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado Maestría en Optimización y Cómputo Aplicado, de la estudiante Rocío Diego Celis, con matrícula 10010396, con el título **DOS ENFOQUES METAHEURÍSTICOS PARA RESOLVER EL PROBLEMA DE IMPRESIÓN DE ETIQUETAS** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
*Por una humanidad culta*  
*Una universidad de excelencia*

**Dr. José Crispín Zavala Díaz**  
**Profesor- investigador**  
**Facultad de Contaduría, Administración e Informática**