INSTITUTO TECNOLÓGICO
DE CD. MADERO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

POR MI PATRIA
Y POR MI BIEN

Maestría en Ciencias en Ciencias de la Computación

OPCIÓN I.- Tesis Profesional

"Análisis experimental de la superficie asociada al espacio de soluciones del Problema de Ordenamiento Lineal con Costos Acumulados (LOPCC)"

Presenta:
I.S.C. Juan Francisco Rocha Aguilar

N° Control:
G04070702

Director de Tesis:
Dr. Héctor Joaquín Fraire Huacuja

Co-director de Tesis:
MC. Jesús David Terán Villanueva

CD. MADERO, TAMPS. MÉXICO Abril 2011.

SEP Institutos Tecnológicos

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

Cd. Madero, Tamps; a 22 de Marzo de 2011

OFICIO No.: U5. 112/11
AREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

C. JUAN FRANCISCO ROCHA AGUILAR
P R E S E N T E

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

"ANÁLISIS EXPERIMENTAL DE LA SUPERFICIE ASOCIADA AL ESPACIO DE SOLUCIONES DEL PROBLEMA DE ORDENAMIENTO LINEAL CON COSTOS ACUMULADOS (LOPCC)"

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

A T E N T A M E N T E
"Por mi Patria y por mi Bien"

Ma. Yolanda Chávez Cinco
M. P. MARÍA YOLANDA CHÁVEZ CINCO
JEFA DE LA DIVISIÓN

ESTADOS UNIDOS MEXICANOS
S.E.P.
DIVISION DE ESTUDIOS
DE POSGRADO E
INVESTIGACION
ITCM

c.c.p.: Archivo

MYCHC 'NLCO 'avgc*

Ave. 10. De Mayo y Sor Juana I. De la Cruz, Col. Los Mangos, C.P. 89440 Cd. Madero, Tam.
Tels. (833) 3 57 48 20, Fax: (833) 3 57 48 20, Ext. 1002, email: itcm@itcm.edu.mx
www.itcm.edu.mx

Resumen

En este trabajo se aborda el Problema de Ordenamiento Lineal con Costos Acumulados (LOPCC). Bertacco demostró que LOPCC es un problema NP-duro [1]. Se realiza el primer estudio de la estructura y la superficie de búsqueda de las instancias del Problema de Ordenamiento Lineal con Costos Acumulados. Esto permite caracterizar las necesidades de intensificación y diversificación de un algoritmo. En base a estos estudios fue posible observar que la rugosidad de la superficie de búsqueda de una instancia es independiente del algoritmo utilizado para generar las soluciones candidatas.

Para el Problema de Ordenamiento Lineal (LOP) se realizó una actualización del estudio de la estructura y la superficie de búsqueda realizado por Schiavinotto [2], para incluir todas las instancias reportadas en [3].

Uno de los trabajos futuros más importantes identificados en este proyecto es realizar un estudio sobre la relación que tiene un algoritmo de solución de un problema con la geometría de la superficie de búsqueda de las instancias, el cual incluya diversos problemas y algoritmos.

Índice general

Índice de tablas

Índice de figuras

CAPÍTULO 1

Introducción

En este capítulo se describen los antecedentes y la formulación del problema de investigación, los objetivos, alcances, limitaciones y metodología de la tesis.

1.1 Contexto del problema

El problema que se aborda en este trabajo de investigación está ubicado en el contexto de las telecomunicaciones, más específicamente en las comunicaciones inalámbricas entre celulares. En este contexto es necesario que los celulares tengan comunicación de forma constante con alguna estación base. Con el fin de distinguir entre las señales de los diferentes celulares, el Estándar Universal de las Telecomunicaciones Móviles adoptó la técnica de División de Código para Múltiple Acceso, mediante la cual cada celular es identificado por un código específico. En el proceso de comunicación entre los celulares móviles y la estación base, los celulares se interfieren unos con otros. Por lo tanto es necesario mantener la interferencia por debajo de un nivel aceptable. Una técnica que se usa para reducir la interferencia entre celulares es llamada Cancelación de Interferencia Sucesiva, por sus siglas en inglés (SIC). De acuerdo a la SIC, las señales de los celulares son detectadas en las estaciones base de acuerdo a un orden predeterminado. Después de cada detección, el enlace radial con el celular detectado es eliminado. Esto permite reducir el nivel de interferencia sobre los celulares aún no detectados, mejorando su detección [1].

Un problema relevante de la técnica de SIC es la elección del orden de detección de los celulares. Generalmente, los usuarios son ordenados de acuerdo al nivel de potencia de emisión (α_i) en orden decreciente, sin embargo, es posible obtener un mejor funcionamiento considerando también el nivel de interferencia entre los usuarios. Un segundo problema es la determinación de la potencia de transmisión de cada celular. Una mayor potencia permite que el celular sea detectado mas fácilmente, genera un alto nivel de interferencia sobre los otros celulares y la carga de la batería dura menos tiempo. Una menor potencia de transmisión hace más difícil la detección del celular, genera menos interferencia y la carga de la batería tiene una mayor duración. Restricciones físicas y la reglamentación de comunicaciones vigente imponen un límite superior (U) sobre la potencia de transmisión de los celulares.

La selección del orden de cancelación y el nivel de potencia de emisión, deben garantizar la detección confiable de todos los teléfonos celulares. Una detección confiable está garantizada cuando la relación de potencia-ruido (SNIR) es igual a un valor determinado Γ. Para un receptor en SIC, el SNIR está relacionado con la potencia de la interferencia generada por el usuario i sobre el usuario j, denotada por ρ_{ij}. En particular, en la detección de un usuario k_p el valor de SNIR está dada por [Bertacco, 2004]:

$$SNIR(P) = \Gamma = \frac{\alpha_{k_p}\rho_{k_p k_p}}{N_O\sqrt{\rho_{k_p k_p}} + \sum_{i\in u_p}\alpha_i N_S \rho_{ik_p}} \tag{1.1}$$

donde:

N_O = Intensidad del ruido.

N_S = Factor de dispersión.

$u_p = \{k_{p+1}, k_{p+2}, ..., k_n\}$ grupo de usuarios no detectados que interfieren al usuario p en el momento de ser detectado.

Al utilizar SIC, se tiene el problema de encontrar el orden de detección que minimíce la potencia de transmisión utilizada por los celulares. Este es conocido como Problema de Optimización Conjunta de la Potencia y la Recepción (JOPCO), el cual fue originalmente planteado por Bertacco en [1]. Formalmente el problema se describe de la manera siguiente:

Problema JOPCO

Dados
$\{1, 2, ..., n\}$: un conjunto de usuarios,
α_i: potencia de transmisión del usuario i,
ρ_{ij}: nivel de interferencia entre los usuarios i y j,
N_O: intensidad del ruido,
N_S: factor de propagación,
U: cantidad máxima de potencia permitida y
Γ: nivel de potencia deseado;

el problema consiste en encontrar una permutación para la detección de usuarios

$$K =< k_1, k_2, ..., k_n >$$

tal que minimice la potencia de transmisión

$$\pi = \sum_{i=1}^{n} \alpha_{k_i} \tag{1.2}$$

bajo las siguientes restricciones:

$$\Gamma = \frac{\alpha_{k_i}\rho_{k_ik_i}}{N_O\sqrt{\rho_{k_ik_i}} + \sum_{l\in u_i}\alpha_l N_S \rho_{lk_i}} \qquad \text{para } i = 1, 2, ..., n \tag{1.3}$$

$$\alpha_{k_i} \leq U \tag{1.4}$$

Por otra parte, JOPCO puede ser formulado de una manera diferente. Si se despeja α_{k_i} de la ecuación 1.3 y se separa en dos términos, se tiene que:

$$\alpha_{k_i} = \frac{\Gamma N_O\sqrt{\rho_{k_ik_i}} + \Gamma N_S \sum_{l\in u_i}\alpha_l\rho_{lk_i}}{\rho_{k_ik_i}} = \frac{\Gamma N_O\sqrt{\rho_{k_ik_i}}}{\rho_{k_ik_i}} + \sum_{l\in u_i}\frac{\Gamma N_S\rho_{lk_i}}{\rho_{k_ik_i}}\alpha_l \tag{1.5}$$

Si se definen:

$$d_{k_i} = \frac{\Gamma N_O\sqrt{\rho_{k_ik_i}}}{\rho_{k_ik_i}} \tag{1.6}$$

y

$$c_{k_il} = \frac{\Gamma N_S\rho_{lk_i}}{\rho_{k_ik_i}}. \tag{1.7}$$

Con estas definiciones la ecuación 1.3 se transforma en la siguiente expresión recursiva:

$$\alpha_{k_i} = d_{k_i} + \sum_{l\in u_i} c_{k_il}\alpha_l \tag{1.8}$$

donde u_i es el conjunto de usuarios aún no detectados que interfieren al usuario i en el momento de ser detectado.

1.2 Problema de Ordenamiento Lineal con Costos Acumulados LOPCC

1.2.1 Formulación del problema

Problema de Ordenamiento Lineal con Costos Acumulados

Dados $G = (V, A)$: un grafo completo con vertices V y aristas A,
d_i: costo no negativo del nodo i,
c_{ij}: costo no negativo de la arista que une los vertices i y j.

El Problema de Ordenamiento Lineal con Costos Acumulados LOPCC consiste en determinar una permutación de los vertices del grafo $P = (p_1, p_2, ..., p_{n-1}, p_n)$ que minimice la función objetivo

$$LOPCC(P) = \sum_{i=n}^{1} \alpha_{p_i}$$

sujeta a las siguientes restricciones:

$$\alpha_{p_i} = d_{p_i} + \sum_{j=i+1}^{n} c_{p_i p_j} \alpha_{p_j} \qquad \text{para} \quad i = n, n-1, ..., 1$$

1.2.2 Cálculo de la función objetivo

Ahora se muestra un ejemplo del cálculo de la función objetivo del problema LOPCC para dos permutaciones dadas.

Para la permutación P_1, la matriz de costos C_1 y el vector de pesos de los nodos D_1 siguientes:

$$P_1 = (1,2,3) C_1 = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 0 & 6 & 5 \\ 1 & 0 & 4 \\ 2 & 3 & 0 \end{pmatrix} \end{array} D_1 = \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1.1 & 2.2 & 3.3 \end{pmatrix} \end{array}$$

el valor objetivo esta dado por:

$C_{LOPCC}(P_1) = \alpha_3 + \alpha_2 + \alpha_1$

Donde el cálculo de los valores de las α's se debe realizar en el orden inverso de la permutación, como se muestra a continuación.

$\alpha_3 = d_3 = 3.3$

$\alpha_2 = d_2 + c_{23}\alpha_3 = 2.2 + 4 * 3.3 = 15.4$

$\alpha_1 = d_1 + c_{12}\alpha_2 + c_{13}\alpha_3 = 1.1 + 6 * 15.4 + 5 * 3.3 = 110$

$C_{LOPCC}(P_2) = 3.3 + 15.4 + 110 = 128.7$

Como se puede observar, en el cálculo de las α's se incluyen todos los elementos del triángulo superior de la matriz C_1.

Ahora si se considera la permutación $P_2 = (1,3,2)$, su valor objetivo se puede calcular de dos formas distintas. Una consiste en usar la definición de las α's y la matriz de costos original C_1 y el vector de pesos original de los nodos D_1.

$C_{LOPCC}(P_1) = \alpha_2 + \alpha_3 + \alpha_1$

Donde el cálculo de los valores de las α's se debe realizar en el orden inverso de la permutación $P_2 = (1,3,2)$, como se muestra a continuación.

$\alpha_2 = d_2 = 2.2$

$\alpha_3 = d_3 + c_{32}\alpha_2 = 3.3 + 3 * 2.2 = 9.9$

$\alpha_1 = d_1 + c_{12}\alpha_2 + c_{13}\alpha_3 = 1.1 + 6 * 2.2 + 5 * 9.9 = 63.8$

$C_{LOPCC}(P_2) = 2.2 + 9.9 + 63.8 = 75.9$

Como se observa el valor de la función objetivo se reduce para esta permutación. Por otra parte, en este caso los costos considerados no son exclusivamente los del triángulo superior de la matriz de costos original. Sin embargo, si se intercambian las columnas y las filas de la matriz original para que correspondan con el orden de la permutación, se generan una matriz de costos permutada C' y un vector de pesos permutado D'.

Matriz original:

$$C_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 6 & 5 \\ 1 & 0 & 4 \\ 2 & 3 & 0 \end{pmatrix} \end{array}$$

Matriz que se obtiene al intercambiar las columnas 2 y 3:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 2 \end{array} \\ \begin{pmatrix} 0 & 5 & 6 \\ 1 & 4 & 0 \\ 2 & 0 & 3 \end{pmatrix} \end{array}$$

Matriz que se obtiene al intercambiar filas 2 y 3:

$$\begin{array}{c} \\ 1 \\ 3 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 2 \end{array} \\ \begin{pmatrix} 0 & 5 & 6 \\ 2 & 0 & 3 \\ 1 & 4 & 0 \end{pmatrix} \end{array}$$

Matriz de costos permutada y vector de pesos permutado:

$$C' = \begin{array}{c} \\ 1 \\ 3 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 2 \end{array} \\ \begin{pmatrix} 0 & 5 & 6 \\ 2 & 0 & 3 \\ 1 & 4 & 0 \end{pmatrix} \end{array} D' = \begin{array}{c} \begin{array}{ccc} 1 & 3 & 2 \end{array} \\ \begin{pmatrix} 1.1 & 3.3 & 2.2 \end{pmatrix} \end{array}$$

Ahora si se calcula el valor objetivo de la permutación $P_2 = (1, 3, 2)$, utilizando la matriz y el vector permutados.

$C_{LOPCC}(P_1) = \alpha_2 + \alpha_3 + \alpha_1$

donde

$\alpha_2 = d'_2 = 2.2$

$\alpha_3 = d'_3 + c'_{32}\alpha_2 = 3.3 + 3 * 2.2 = 9.9$

$\alpha_1 = d'_1 + c'_{13}\alpha_3 + c'_{12}\alpha_2 = 1.1 + 6 * 2.2 + 5 * 9.9 = 63.8$

$C_{LOPCC}(P_2) = 2.2 + 9.9 + 63.8 = 75.9$

Se observa que los costos involucrados corresponden con los costos del triángulo superior de la matriz permutada. Por lo tanto el problema de LOPCC consiste en determinar una permutación de las filas y las columnas de la matriz de costos dada, de tal manera que se minimice el valor de la función objetivo. Sin embargo, como el cálcluo de la matriz y el vector permutados es muy costoso, es más económico calcular el valor objetivo directamente a partir de la matriz original, como se mostró previamente.

1.2.3 Complejidad del problema

Bertacco probó que el Problema de Ordenamiento Lineal con Costos acumulados es NP-duro [1]. Para demostrarlo se prueba que el problema del ciclo hamiltoniano (HC) se transforma polinomiálmente a LOPCC.

El problema de HC consiste en determinar si un grafo dado contiene un ciclo que visite todos los vértices sin pasar dos veces por el mismo vértice.

1.3 Preguntas de investigación

En el análisis del estado del arte se identificaron varias areas de oportunidad. Con base en estas areas se propone un conjunto de preguntas de investigación en relación a LOPCC. Las principales preguntas que se plantean son las siguientes:

- ¿Es posible balancear la intensificación y diversificación para el algoritmo metaheurístico de búsqueda Tabú presentado por Laguna?
- ¿Es posible implementar diferentes técnicas de reencadenamiento de trayectorias en búsqueda de mejores soluciones como la implementada por Laguna?
- ¿Es posible investigar e implementar una buena estrategia para la construcción de soluciones iniciales?
- ¿Es posible implementar un algoritmo memético, el cual cuente con un procedimiento que permita acercarse a un óptimo local, antes de realizar la búsqueda local?
- ¿Es posible llevar a cabo un análisis de la estructura y de la superficie de búsqueda asociada a las instacias de LOPCC?

En este proyecto de tesis se propone ampliar el conocimiento acerca de la última cuestión, debido a que hasta el momento no existe ningún tipo de estudio de este tipo para LOPCC.

1.4 Objetivos

1.4.1 Objetivo general

Realizar el análisis de la estructura y de la superficie de búsqueda para las instancias del Problema de Ordenamiento Lineal con Costos Acumulados, reportadas en la literatura.

1.4.2 Objetivos específicos

- Revisar la literatura relacionada con la superficie de soluciones.

- Analizar algoritmos y códigos disponibles (metaheurísticos): búsqueda Tabú LOP, búsqueda Local Iterada (ILS) LOP, búsqueda dispersa LOP y búsqueda Tabú LOPCC.
- Analizar, diseñar e implementar un algoritmo memético y un algoritmo de búsqueda Local Iterada para LOPCC.
- Analizar la estructura de las instancias de LOP y LOPCC.
- Realizar un estudio de la correlación de la superficie de búsqueda de LOP y LOPCC.
- Realizar un estudio de la distancia de aptitud de la superficie de búsqueda de LOPCC.

1.5 Alcances y limitaciones

- En el análisis de la estructura de las instancias solo se consideran la dispersión, el coeficiente de variación y el sesgo.
- En el análisis de la superficie de búsqueda solo se considera la longitud de correlación y la distancia de aptitud.
- En el estudio se incluyen todas las instancias reportadas en [4].

CAPÍTULO 2

Marco teórico

En este capítulo se describe el marco teórico en que se fundamenta este trabajo. Se presentan el planteamiento formal de LOPCC y LOP, las instancias utilizadas y los fundamentos para análisis de la estructura y de la superficie de búsqueda.

2.1 Problema de Ordenamiento Lineal (LOP)

LOP es un problema de interés que ha sido documentado en la literatura [5]. Una aplicación conocidos de LOP es la triangulación de las matrices de insumo-producto, para determinar la estabilidad de una economía. Otro aplicación es el de la estratificación en la arqueología, donde el problema de ordenamiento lineal (LOP) es utilizado para encontrar el orden cronológico más probable de las muestras que se encuentran en diferentes sitios [2]. La definición formal de LOP es presentada a continuación:

Dada una matriz de pesos E $=\{e_{ij}\}$, de tamaño $m * m$, LOP consiste en encontrar una permutación $P = (p_1, p_2 ... p_n)$ de las columnas y las filas de E, de tal manera que se maximice la suma de los pesos ubicados en el triángulo superior de la matriz. En términos matemáticos se busca maximizar [5]:

$$C_E(p) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (e_{p_i p_j})$$

2.2 Problema de Ordenamiento Lineal con Costos Acumulados (LOPCC)

Formulación del problema:

Dados $G = (V, A)$: un grafo completo con vertices V y aristas A,
d_i: costo no negativo del nodo i,
c_{ij}: costo no negativo de la arista que une los vertices i y j.

El Problema de Ordenamiento Lineal con Costos Acumulados LOPCC consiste en determinar una permutación de los vertices del grafo $P = (p_1, p_2, ..., p_{n-1}, p_n)$ que minimice la función objetivo

$$LOPCC(P) = \sum_{i=n}^{1} \alpha_{p_i}$$

sujeta a las siguientes restricciones:

$$\alpha_{p_i} = d_{p_i} + \sum_{j=i+1}^{n} c_{p_i p_j} \alpha_{p_j} \qquad \text{para} \quad i = n, n-1, ..., 1$$

2.3 Instancias empleadas

2.3.1 Instancias para LOP

Para realizar el estudio de la estructura de las instancias y de la longitud de correlación de la superficie de búsqueda para el problema LOP se utilizaron las instancias reportadas en [3]. Una descripción de dichas instancias es la siguiente:

- **LOLIB**. Instancias provenientes de casos reales de Problemas de Ordenamiento Lineal. Estas instancias fueron generadas a partir de tablas de entrada/salida de diversas fuentes [3]. En total este grupo está formado por 50 instancias.
- **SGB**. Instancias tomadas de Stanford GraphBase. Estas instancias fueron formadas a partir de tablas de entrada/salida de los sectores de la economía de los Estados Unidos. El conjunto tiene un total de 25 casos de tamaño 75.
- **Random AI**. Instancias generadas con una distribución uniforme en el rango de 0 a 100. Fueron propuestas en [6] y generadas en [7]. Este grupo esta formado con 25 instancias de tamaño 100, 25 de tamaño 150 y 25 de tamaño 200. Posteriormente se incluyeron 25 instancias adicionales de tamaño 500 [3].
- **Random AII**. Instancias generadas a partir de contar el número de veces que un sector aparece en una posición más alta que otro en una serie de permutaciones generadas al azar. Los tamaños de este tipo de instancias son de 100, 150 y 200, con 25 instancias de cada tamaño.
- **Random B**. En estas instancias los elementos sobre la diagonal se generan aleatoriamente de manera uniforme en el intervalo $([0, U_1])$ y los elementos bajo la diagonal se generan en el intervalo $([0, U_2])$ donde U_1 y U_2 son numeros enteros generados uniformemente de tal manera que $U_1 \geq U_2$.

- **MB**. Instancias generadas por Mitchel y Borchers. Ellos generan una matriz, para la cual asignan valores aleatorios entre 0 y 99 a los elementos localizados en el triángulo superior y entre 0 y 39 a los restantes. De este tipo de instancias se tienen disponibles cinco de tamaño 100, diez de tamaño 150, diez de tamaño 200 y 5 de tamaño 250

- **XLOLIB**. Instancias creadas en [2] a partir de las instancias LOLIB para obtener problemas más grandes. Por cada instancia de LOLIB se generaron dos casos grandes, uno de tamaño 150 y otro de tamaño 250. Como resultado se obtuvieron 49 instancias de tamaño 150 y 49 instancias de tamaño 250. Por lo tanto, este grupo se compone de 98 casos. De las 98 instancias fueron removidas 20, debido a que los valores de entrada de estas 20 instancias resultaron ser muy grandes para ser representados con un entero de 4 Bytes, por lo que se redujeron a 78 instancias.

- **Specials**. Las instancias Specials están formadas por casos reportados en diferentes experimentos computacionales. Los casos econ, con 36 casos, fueron generados a partir de la matriz usa79. Los casos ATP fueron creados a partir de los resultados de los torneos de tenis de la ATP en 1993/1994. Los casos Paley se derivan a partir de una definición algebraica.

2.3.2 Instancias para LOPCC

Las instancias de LOPCC están organizadas por grupos de acuerdo a su origen. Las instancias utilizadas en este trabajo fueron reportadas en [4] y son las siguientes:

- **UMTS**. Instancias aportadas por el grupo de telecomunicaciones de la escuela de ingeniería de la Universidad de Padova. Están relacionadas con el problema de la optimización del orden de detección en redes UMTS. Este grupo esta formado por cuatro grupos de 25 instancias de tamaño 16.

- **Random**. Instancias generadas mediante una construcción uniforme propuesta en [?]. Este grupo esta formado por 25 instancias de tamaño 35, 25 de tamaño 100 y 25 de tamaño 150.

- **LOLIB**. Estas instancias provienen de las tablas de entrada-salida de los sectores de economía de Europa. Se cuenta con 49 instancias, 31 de tamaño 44, 4 de tamaño 50, 4 de tamaño 56 y 3 de tamaño 60.

2.4 Análisis de la estructura de las instancias

El análisis de la estructura de las instancias provee información estadística de la distribución de los datos en las estructuras de datos asociadas a cada una de las instancias. Los indicadores que se utilizan son los siguientes:

- **Dispersión (Sp)**. Se define como el porcentaje de ceros en las estructuras de las instancias del problema [2].

- **Coeficiente de variación (VC)**. Se define como la razón entre la media y la desviación estándar de los datos en las estructuras asociadas a las instancias ($\frac{\sigma}{\mu}$) [2, 8].

- **Sesgo (Sk)**. Indica el grado de asimetría de la distribución de los datos de las instancias del problema. Se define como el tercer momento de la media normalizada por la desviación standard [2, 9]. Su formulación es la siguiente:

$$SK = E\left(\frac{X-\mu}{\sigma}\right) = \frac{E[(X-\mu)^3]}{E[(X-\mu)^2]^{3/2}} = \frac{\frac{1}{2}\sum_{i=1}^{n}(x_i-\overline{x})^3}{\left(\frac{1}{2}\sum_{i=1}^{n}(x_i-\overline{x})^2\right)^{3/2}} \tag{2.1}$$

2.5 Análisis de superficie de búsqueda

La superficie de búsqueda generada por un algoritmo al resolver una instancia dada esta determinada por las soluciones factibles visitadas y sus valores objetivo. El análisis de la superficie de búsqueda permite establecer una relación entre las características de la geometría de la superficie y las de los algoritmos de solución de las instancias de un problema [10, 11]. De manea formal, el espacio de soluciones de LOPCC se describe como una tripleta $< \Pi(n), f, d >$, donde Π es el conjunto de todas las permutaciones de n enteros, f es la función objetivo del problema y d es una medida de distancia.

Para realizar el análisis de la superficie de búsqueda se consideran 2 características: la longitud de la correlación y la distancia de aptitud. Estos conceptos se definirán detalladamente en las siguientes secciones.

2.5.1 Distancia entre permutaciones

Existen diferentes métricas para determinar la distancia entre permutaciones [12]. En este trabajo se usa la distancia definida en [13] la cual se describe continuación:

Distancia entre permutaciones
Sean
p y p' permutaciones de n enteros,
n_{ij} el número de veces que $i \in p$ está antes de $j \in p$ en ambas permutaciones,
$dmax = n(n-1)/2$, y $N = \sum_{i,j \in p} n_{ij}$.

La distancia entre las permutaciones p y p' se define como:
$d(p,p')=dmax$ - N

$dmax$ corresponde a la distancia máxima entre las permutaciones. El siguiente ejemplo muestra como calcular la distancia entre dos permutaciones de tamaño $n = 4$.

Para este ejemplo se consideran las permutaciones $p = (1, 2, 3, 4)$ y $p' = (4, 3, 2, 1)$.

Para estas dos permutaciones la cantidad de veces que i está antes que j en ambas permutaciones es igual a cero, por lo que la distancia está dada directamente por $n(n-1)/2$.

$dmax = n(n-1)/2$
$= 4(4-1)/2$
$= 4(3)/2$
$= 12/2$
$= 6$

$d(p,p')=dmax \; - 0 = 6$

2.5.2 Análisis de correlación de superficie (rugosidad)

El análisis de correlación de superficie es usado para estudiar la rugosidad de la superficie de búsqueda. si hay poca correlación entre los valores objetivos asociados a la superficie, se considera que es rugosa. Dada una serie de tiempo formada con los valores objetivos de las soluciones generadas $\{f(p_i), i = 1, \ldots, m\}$, la autocorrelación se define como la correlación entre pares de soluciones separadas por s pasos durante una caminata aleatoria de tamaño m a través del área de soluciones [14].

$$r(s) = \frac{1}{\sigma^2(f)(m-s)} \sum_{i=1}^{m-s} (f(p_i) - \overline{f})(f(p_{i+s}) - \overline{f}) \tag{2.2}$$

Donde $\sigma^2(f)$ y $\overline{f}$ son la varianza y la media de la serie. La longitud de correlación de la superficie de búsqueda se define como $l = -\frac{1}{\ln(|r(1)|)}$ donde $|r(1)| \neq 0$, [15]. Entre menor es el valor de l, más rugosa será la superficie de búsqueda [16].

2.5.3 Análisis de la distancia de aptitud

Mediante este estudio se analiza la distribución y la ubicación relativa de óptimos locales en relación al mejor conocido, lo cual da idea de la calidad de las soluciones que obtenemos mediante el uso de búsquedas locales [2].

Este estudio se realiza mediante el índice de la correlación de distancia de aptitud, el cual se obtiene de la siguiente manera:

Dada una muestra de m soluciones $\pi_1, ..., \pi_m$ con un conjunto de pares asociados $(f_1, d_1), \ldots, (f_m, d_m)$, donde la aptitud f_i corresponde al valor objetivo de la solución p_i y d_i es la distancia de p_i hacia el óptimo global o mejor conocido, la correlación de la distancia de aptitud ρ se define como:

$$\rho = \frac{Cov(f,d)}{\sigma(f) \cdot \sigma(d)}$$

En la cual

$$Cov(f,d) = \frac{1}{m-1} \sum_{i=1}^{m} (f_i - \overline{f})(d_i - \overline{d})$$

y

$$\sigma(f) = \sqrt{\tfrac{1}{m-1} \textstyle\sum_{i=1}^{m} (f_i - \overline{f})^2} \qquad \sigma(d) = \sqrt{\tfrac{1}{m-1} \textstyle\sum_{i=1}^{m} (d_i - \overline{d})^2}$$

donde $\overline{f}$, $\overline{d}$ son las medias de los conjuntos $F = f_1, ..., f_m$ y $D = d_1, ..., d_m$ respectivamente, $Cov(f,d)$ es la covarianza entre f y d, mientras que $\sigma(f)$ y $\sigma(d)$ son las desviaciones estándar de F y D, respectivamente. Para calcular la desviación estándar se calcula la distancia hacia el óptimo global. Un valor más positivo de ρ indica una buena calidad promedio de las soluciones consideradas.

2.5.4 Vecindades

En los problemas de optimización combinatoria, el espacio de soluciones S se define como el conjunto de todas las soluciones factibles o candidatas del problema [17]. Dado S y $x \in S$, una vecindad $N(x)$ esta formada por todas las soluciones $y \in S$ que se obtienen aplicando una transformación o movimiento a la configuración de x. En cierto sentido $N(x)$ esta formada por todas las soluciones candidatas cercanas a x.

Movimiento de inserción

Dado el ordenamiento $p = (p_1, p_2, ..., p_m)$, $INSERT\_MOVE(p_j, i)$ es un movimiento de inserción que consiste en eliminar el elemento p_j de su posición actual j para insertarlo en la posición i. Esta transformación da como resultado el ordenamiento p', como se muestra a continuación:

$$p' = \begin{cases} (p_1, ..., p_{i-1}, p_j, p_i, ..., p_{j-1}, p_{j+1}, ..., p_m) & \text{para } i < j \\ (p_1, ..., p_{j-1}, p_{j+1}, ..., p_i, p_j, p_{i+1}, ..., p_m) & \text{para } j < i \end{cases}$$

Movimiento de intercambio

Dado el ordenamiento $p = (p_1, p_2, ..., p_m)$, $INTERCHANGE(i, j)$ es un movimiento de intercambio que consiste en intercambiar en p los elementos p_i y p_j. Esta transformación da como resultado un nuevo ordenamiento p', como se muestra a continuación:

$p' = \ (p_1, ..., p_{i-1}, p_j, p_{i+1}, ..., p_{j-1}, p_i, p_{j+1}, ..., p_m)\ .$

Vecindad N_1 [5]

La vecindad N_1 esta formada por todos los ordenamientos p' que se obtienen de p, al insertar cada uno de sus elementos en la posición siguiente.

$$N_1 = \{\ p' \in S : p' = INSERT\_MOVE(p_j, i), \text{ para } j = 1, 2, ..., m-1 \text{ y } i = j+1\ \}$$

Ejemplo de vecindad N_1

Dada la permutación $p = (2, 6, 1, 4, 3, 5)$, la Figura 2.1 muestra todos los elementos de la vecindad $N_1(p)$.

$$\begin{aligned}
p'_1 &= INSERT_MOVE(2, 2) = (6, 2, 1, 4, 3, 5) \\
p'_2 &= INSERT_MOVE(6, 3) = (2, 1, 6, 4, 3, 5) \\
p'_3 &= INSERT_MOVE(1, 4) = (2, 6, 4, 1, 3, 5) \\
p'_4 &= INSERT_MOVE(4, 5) = (2, 6, 1, 3, 4, 5) \\
p'_5 &= INSERT_MOVE(3, 5) = (2, 6, 1, 4, 5, 3)
\end{aligned}$$

Figura 2.1: Vecindad N_1

Vecindad N_2 [5]

La vecindad N_2 esta formada por todos los ordenamientos p' que se obtienen de p, al insertar cada uno de sus elementos en otra posición diferente de la posición actual del elemento insertado.

$$N_2 = \{\ p' \in S : p' = INSERT\_MOVE(p_j, i), \text{ para } \\ j = 1, 2, ..., m \text{ y } i = 1, 2, ..., j-1, j+1, ..., m\ \}$$

Ejemplo de vecindad N_2

Dada la permutación $p = (2, 6, 1, 4, 3, 5)$, las Figuras 2.2 2.3 muestran los elementos de la vecindad $N_2(p)$. Se observa que los vecinos del 1 al 15 corresponden a movimientos de inserción hacia adelante y los movimientos del 16 al 30 corresponden a inserciones hacia atrás.

Vecindad N_x [2]

La vecindad N_x esta formada por todos los ordenamientos p' que se obtienen de p, al intercambiar todos los posibles pares de elementos.

$$N_2 = \{\ p' \in S : p' = INTERCHANGE(i, j), \text{ para } i = 1, 2, ..., m,\ j = 1, 2, ..., m \text{ e } i \neq j\ \}$$

Ejemplo de vecindad N_x

Dada la permutación $p = (2, 6, 1, 4, 3, 5)$, la Figura 2.4 muestra todos los elementos de la vecindad $N_x(p)$.

Como se puede observar de las 3 vecindades definidas, la vecindad de inserción N_2 es la que produce una mayor cantidad de soluciones vecinas. En general la cardinalidad de la vecindad N_1 es $n-1$, la de la vecindad N_2 es $(n-1)n$ y la cardinalidad de la vecindad N_x es $((n-1)n)/2$

$$
\begin{aligned}
p'_1 &= INSERT\_MOVE(2,2) = (6,2,1,4,3,5)\\
p'_2 &= INSERT\_MOVE(2,3) = (6,1,2,4,3,5)\\
p'_3 &= INSERT\_MOVE(2,4) = (6,1,4,2,3,5)\\
p'_4 &= INSERT\_MOVE(2,5) = (6,1,4,3,2,5)\\
p'_5 &= INSERT\_MOVE(2,6) = (6,1,4,3,5,2)\\
p'_6 &= INSERT\_MOVE(6,3) = (2,1,6,4,3,5)\\
p'_7 &= INSERT\_MOVE(6,4) = (2,1,4,6,3,5)\\
p'_8 &= INSERT\_MOVE(6,5) = (2,1,4,3,6,5)\\
p'_9 &= INSERT\_MOVE(6,6) = (2,1,4,3,5,6)\\
p'_{10} &= INSERT\_MOVE(1,4) = (2,6,4,1,3,5)\\
p'_{11} &= INSERT\_MOVE(1,5) = (2,6,4,3,1,5)\\
p'_{12} &= INSERT\_MOVE(1,6) = (2,6,4,3,5,1)\\
p'_{13} &= INSERT\_MOVE(4,5) = (2,6,1,3,4,5)\\
p'_{14} &= INSERT\_MOVE(4,6) = (2,6,1,3,5,4)\\
p'_{15} &= INSERT\_MOVE(3,6) = (2,6,1,4,5,3)
\end{aligned}
$$

Figura 2.2: Vecindad N_2

$$p'_{16} = INSERT\_MOVE(5,5) = (2,6,1,4,5,3)$$
$$p'_{17} = INSERT\_MOVE(5,4) = (2,6,1,5,4,3)$$
$$p'_{18} = INSERT\_MOVE(5,3) = (2,6,5,1,4,3)$$
$$p'_{19} = INSERT\_MOVE(5,2) = (2,5,6,1,4,3)$$
$$p'_{20} = INSERT\_MOVE(5,1) = (5,2,6,1,4,3)$$
$$p'_{21} = INSERT\_MOVE(3,4) = (2,6,1,3,4,5)$$
$$p'_{22} = INSERT\_MOVE(3,3) = (2,6,3,1,4,5)$$
$$p'_{23} = INSERT\_MOVE(3,2) = (2,3,6,1,4,5)$$
$$p'_{24} = INSERT\_MOVE(3,1) = (3,2,6,1,4,5)$$
$$p'_{25} = INSERT\_MOVE(4,3) = (2,6,4,1,3,5)$$
$$p'_{26} = INSERT\_MOVE(4,2) = (2,4,6,1,3,5)$$
$$p'_{27} = INSERT\_MOVE(4,1) = (4,2,6,1,3,5)$$
$$p'_{28} = INSERT\_MOVE(1,2) = (2,1,6,4,3,5)$$
$$p'_{29} = INSERT\_MOVE(1,1) = (1,2,6,4,3,5)$$
$$p'_{30} = INSERT\_MOVE(6,1) = (6,2,1,4,3,5)$$

Figura 2.3: Vecindad N_2

$$p'_1 = INTERCHANGE(1,2) = (2,6,1,4,3,5)$$
$$p'_2 = INSERT\_MOVE(1,3) = (1,6,2,4,3,5)$$
$$p'_3 = INSERT\_MOVE(1,4) = (4,6,1,2,3,5)$$
$$p'_4 = INSERT\_MOVE(1,5) = (3,6,1,4,2,5)$$
$$p'_5 = INSERT\_MOVE(1,6) = (5,6,1,4,3,2)$$
$$p'_6 = INSERT\_MOVE(2,3) = (2,1,6,4,3,5)$$
$$p'_7 = INSERT\_MOVE(2,4) = (2,4,1,6,3,5)$$
$$p'_8 = INSERT\_MOVE(2,5) = (2,3,1,4,6,5)$$
$$p'_9 = INSERT\_MOVE(2,6) = (2,5,1,4,3,6)$$
$$p'_{10} = INSERT\_MOVE(3,4) = (2,6,4,1,3,5)$$
$$p'_{11} = INSERT\_MOVE(3,5) = (2,6,3,4,1,5)$$
$$p'_{12} = INSERT\_MOVE(3,6) = (2,6,5,4,3,1)$$
$$p'_{13} = INSERT\_MOVE(4,5) = (2,6,1,3,4,5)$$
$$p'_{14} = INSERT\_MOVE(4,6) = (2,6,1,5,3,4)$$
$$p'_{15} = INSERT\_MOVE(5,6) = (2,6,1,4,5,3)$$

Figura 2.4: Vecindad N_x

CAPÍTULO 3

Estado del arte

3.1 LOP

Para el Problema de Ordenamiento Lineal (LOP) existe un gran número de trabajos, dado que tiene un largo tiempo de ser tratado. De lo más destacado en los últimos años se tienen los realizados por Laguna [5] y Schiavinotto [2].

Laguna, en el trabajo "**Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem**", propone un metaheurístico basado en la metodología de búsqueda Tabú. Este algoritmo incorpora estrategias de búsqueda orientadas a la intensificación y diversificación de corto plazo. Además, aplica intensificación adicional mediante reencadenamiento de trayectorias (Path-Relinking) y diversificación de largo plazo. El proceso de búsqueda de este metaheurístico inicia con una solución totalmente aleatoria. La Figura 3.1 muestra la estructura general del algoritmo de búsqueda Tabú de Laguna seguida de una pequeña explicación de cada una de sus estructuras.

Intensificación. En una iteración se escoge al mejor vecino, moviendo un sector j, *SolInicialAleatoria()*

Actualizó la mejor solución encontrada(mejor_global)

While*(!=CantidadIteracionesSinMejora)*

{

Intensificación(mejor_global)

Se verifica si se encontró una mejor solución global y en caso de ser así se reinicia el iterador global

Reencadenamiento de trayectorias()

Diversificación(mejor_global)

Se verifica si se encontró una mejor solución global y en caso de ser así se reinicia el iterador global

DiversificacionaLargoPlazo()

}

Figura 3.1: Estructura básica del algoritmo de búsqueda Tabú LOP

aunque deteriore el valor de la función objetivo. Una vez seleccionado el movimiento, se establece como Tabú activo el sector utilizado, por lo cual no podrá ser seleccionado nuevamente durante un número de iteraciones determinado. La cantidad de veces que es seleccionado un sector j, es acumulado en un contador de frecuencia, el cual será utilizado en la fase de diversificación adicional. La fase de intensificación termina después de un número $MaxInt$ consecutivo sin mejora. Antes de terminar la fase de intensificación se aplica la estrategia $first(N_2)$ para obtener un óptimo local $P^{\#}$. Durante este proceso se almacenan las 4 mejores soluciones encontradas, a las cuales posteriormente se les aplicará reencadenamiento de trayectorias.

Diversificación. En esta fase se selecciona un sector al azar. La probabilidad de que el sector j sea seleccionado es inversamente proporcional a la cantidad de veces que fue seleccionado dicho sector en la fase de intensificación. El sector elegido es ubicado en la mejor posición. La fase de diversificación termina cuando se llega a un número máximo $MaxDiv$ sin mejora. Este proceso de diversificación permite construir una solución que puede ser utilizada como punto de partida en el siguiente ciclo de intensificación. Sin embargo, antes de iniciar este último ciclo a dicha solución se le aplica un mecanismo de diversificación adicional.

Reencadenamiento de trayectorias. En este procedimiento se forman pares con las 4 soluciones elite construidas en la fase de intensificación. A cada par se le aplica un proceso para ir paso a paso de la solución inicial a la solución guía en el par. Básicamente se identifican los elementos que ocupan las mismas posiciones en ambas soluciones y se realizan operaciones de inserción para colocar los elementos diferentes, en la posición que les corresponde en la solución guía. Para cada uno de estos últimos elemento, se genera una nueva solución intermedia entre la solución inicial y la guía. Cada una de las soluciones intermedias que se construyen es evaluada y se actualiza la mejor solución encontrada.

Diversificación a largo plazo.

Este proceso aplica diversificación adicional a la solución generada $p$con el proceso de diversificación previo. Para cada sector i de dicha solución se determina la posición promedio $l(i)$que ocupa en las soluciones elite. Posteriormente cada sector es insertado en la posición complementaria con respecto a la posición promedio. Para cada sector: $INSERT\_MOVE(p, i, m - l(i))$

Schiavinotto, en el trabajo "**The Linear Ordering Problem: Instances, Search Space Analysis and Algorithms**" [2], analiza el espacio de búsqueda de las instancias de LOP. Para este trabajo, toma las instancias usadas hasta ese momento en diferentes investigaciones: LOLIB, SGB, XLOLIB, MBLB Y LMC-LOP. Como primer punto de este estudio se analiza la estructura de las instancias del problema. Los indicadores estadísticos usados en este análisis son la dispersión, el coeficiente de variación y el sesgo. En base a los resultados obtenidos, Schiavinotto, indica la diferencia significante entre las instancias del la vida real (LOLIB, XLOLIB y SGB) y las instancias generadas de manera aleatoria(LMCLOP y MBLB), siendo mucho mayores los indicadores para las instancias de la vida real, lo que sugiere que estas son menos regulares y que la variación de los datos de la matriz de entrada de estas es mucho mayor.

El segundo estudio realizado es el análisis de la longitud de correlación de superficie. Este estudio muestra de acuerdo al promedio, que la variación de la longitud de correlación entre los grupos de instancias es relativamente pequeña. De los grupos analizados, las instancias MBLB presentan el índice mayor de correlación de superficie, lo que sugiere que estas son las mas fáciles, lo cual fue también observado en estudios experimentales. Después de las instancias MBLB, en orden decreciente de longitud de correlación se encuentran las instancias LOLIB, SGB, LMC-LOP y XLOLIB respectivamente.

El ultimo estudio realizado es el análisis de distancia de aptitud. Para este estudio se realiza un número n de búsquedas locales que parten de soluciones aleatorias. Para este trabajo, a las instancias LOLIB se le realizó 13000 búsquedas locales y 1000 búsquedas Locales a los demás grupos de instancias. En este estudio se consideraron diferentes aspectos, como la cantidad de búsquedas que llegaron a óptimos locales distintos y la cantidad de búsquedas que llegaron al óptimo global. De acuerdo a los resultados obtenidos, se observa que la cantidad de óptimos locales distintos encontrados y la cantidad de óptimos locales que llegan al óptimo global varían. Para las instancias LMC-LOP y XLOLIB el total de las búsquedas locales aplicadas llegaron a soluciones distintas y ninguna de ellas llegó al óptimo global. Para las instancias LOLIB el numero de óptimos locales distintos encontrados va de 24 a 13000 con una mediana de 9 400; para las MBLB de 73 a 1000 con una mediana de 965 y finalmente para las instancias SGB de 600 a 1000. Por otro lado, en base a la calidad de los óptimos locales encontrados, Schiavinotto menciona que las instancias MBLB serían las mas fáciles de resolver, concordando las conclusiones obtenidas en el estudio de longitud de correlación y en resultados experimentales con metaheuristicos y algoritmos exactos.

Otro trabajo de interés es el presentado por Rafael Martí, denominado "A Benchmark Library and a Comparison of Heuristic Methods for the Linear Ordering Proble" [3], en el que evalúa el desempeño de una colección de heurísticas y metaheurísticas. Para la evaluación experimental del desempeño utiliza versiones normalizadas de las instancias reportadas en el trabajo anterior y se agregan nuevas instancias.

Como se observa en el estudio de los trabajos relacionados, se requiere realizar el estudio de la estructura y de la superficie de búsqueda para las instancias incorporadas en el conjunto de casos de prueba estándar utilizado en [3].

3.2 LOPCC

Los trabajos enfocados a LOPCC son muy pocos, ya que este problema se originó recientemente. Su planteamiento inicial fue realizado por Bertacco en el 2004 [1]. En su trabajo, además de exponer los fundamentos, propone dos métodos de solución exacta para abordarlo. Los métodos propuestos son un algoritmo de Programación Mixta Entero-Lineal (MIP) y un algoritmo enumerativo.

Giovanni Righini propone un algoritmo enumerativo modificado [18]. Incorpora una nueva restricción al modelo y una nueva política para el ordenamiento de las ramas de cada nodo. A partir de estas modificaciones propone un algoritmo heurístico basado en el truncamiento del algoritmo exacto.

Duarte propone un algoritmo metaheurístico basado en búsqueda Tabú [4]. Para el diseño del metaheurístico considera una vecindad de inserción y un mecanismo eficiente para predecir el costo de una inserción. El algoritmo incorpora un mecanismo para la construcción de la solución inicial que realiza un gran porcentaje de avance hacia el óptimo global. Una vez construida la solución inicial se inicia un ciclo global en el que se realiza en secuencia un proceso de intensificación y uno de diversificación. La intensificación consiste en una búsqueda local que reubica los diferentes elementos de la solución actual, tratando de que mejore su valor objetivo. En este proceso se utiliza una lista Tabú para evitar el uso de los elementos recientemente utilizados durante un cierto número de iteraciones. Las mejores soluciones encontradas durante la intensificación son almacenadas para ser utilizadas en el proceso de diversificación. La diversificación consiste en generar aleatoriamente una nueva solución con una estructura diferente a la de las soluciones elite. Con este propósito, se calcula la frecuencia con que los diferentes elementos de una solución aparecen en las diferentes posiciones de las soluciones elite almacenadas. Posteriormente se genera una nueva solución dando preferencia a las posiciones menos utilizadas para ubicar los diferentes elementos de la nueva solución. Esta nueva solución es utilizada para iniciar el proceso de intensificación en la siguiente iteración. La Figura 5.3 muestra la estructura general de este algoritmo.

ConstructSolInicial()
Actualizó la mejor solución encontrada(p_overall_best)
While*(!=CantidadIteracionesSinMejora)*
{
    Intensificación(p_overall_best)
    Se verifica si se encontró una mejor solución global y en caso de ser así se reinicia el iterador global
    Diversificación(p_overall_best)
    Se verifica si se encontró una mejor solución global y en caso de ser así se reinicia el iterador global
}

Figura 3.2: Estructura básica del algoritmo Tabu LOPCC

Para la evaluación experimental del desempeño se utilizan las instancias UMTS, LOLIB y Random.

Como se puede observar, aún no se ha realizado el análisis de la estructura y de la superficie de búsqueda de las instancias estándar de LOPCC. En este trabajo se propone realizar estos estudios con el fin de aportar nuevo conocimiento que sirva como referencia para el diseño de nuevas soluciones heurísticas.

CAPÍTULO 4

Estudio de la estructura de las instancias

En este capítulo se presenta el estudio de la estructura de las instancias del problema LOP y LOPCC. Aquí se exponen las características de estructura evaluadas (dispersión, coeficiente de variación, y sesgo), así como el algoritmo usado en estas evaluaciones. Para finalizar se muestra un grupo de comentarios en forma de conclusión.

4.1 Descripción del estudio de la estructura de las instancias

En este análisis se obtuvo información estadística de la distribución de los datos en la matriz de costos de cada una de las instancias reportadas en el estado del arte de LOP y LOPCC. Para cada una de estas instancias se calculó su índice de dispersión, coeficiente de variación y su sesgo.

Para realizar las pruebas se utilizó un servidor con un procesador Intel XEON 3.06

GHZ, 4GB de ram y 70 GB de disco duro. El ambiente de programación usado fue Visual Studio 2005 y se usó el compilador Visual C++.

Para esta prueba se capturó el contenido de la matriz de costos de cada una de las instancias para posteriormente obtener las métricas de **dispersión, coeficiente de variación y sesgo**. La Figura 4.1 muestra en términos generales los procesos realizados en esta prueba.

ObtenerTamañoInstancia()
While (ExistanInstancias)
{
    Instancia = NomInstancia()
    π = *LeerDatosdeInstancia(Instancia)*
    Sp= ObtenerDispersión(π)
    VC= ObtenerCoefVariación(π)
    Sk= ObtenerSesgo(π)
}

Figura 4.1: Estructura básica del algoritmo: Análisis de estructura

Como primer punto, en el algoritmo se obtuvo el tamaño de la instancia para poder determinar la cantidad de datos que contiene la matriz de entrada. Posteriormente se implementó una estructura iterativa para controlar la cantidad de instancias a evaluar. Dentro de este ciclo se captura el contenido de cada instancia y se evalúan sus índices estadísticos.

Para el análisis de LOP cada grupo de instancias fue subdividido en base al conocimiento del óptimo de cada instancia. En esta clasificación, las instancias con óptimo conocido se denominan OPT-I y a las instancias cuyo óptimo se desconoce UB-I. Esta clasificación se muestra en la Tabla 4.1.

Tabla 4.1: Agrupación de las instancias LOP en base al conocimiento del óptimo de cada instancia

Grupo	No. Instancias	OPT-I	UB-I
LOLIB (IO)	50	50	-
SGB	25	25	-
RandomAI	100	-	100
RandomAII	75	25	50
RandomB	90	70	20
MB	30	30	-
XLOLIB	78	-	78
Special	36	29	6

Para ambos problemas(LOP y LOPCC) las instancias se clasificaron en varios grupos para obtener información que permita explicar la dificultad de las instancias de cada grupo.

La clasificación usada para LOPCC se muestra en la tabla 4.2

Tabla 4.2: Agrupación de las instancias LOPCC para análisis de estructura

Grupo	No. Instancias
UMTS	75
LOLIB (IO)	49
Random	100

4.2 Estudio de la estructura de las instancias del Problema de Ordenamiento Lineal

Los resultados del estudio de la estructura de las instancias OPT-I de LOP se presentan en la Tabla 4.3, y de las instancias UB-I de LOP en la Tabla 4.4. La primera columna de las tablas contiene el grupo de instancias que corresponde a los índices mostrados. La segunda muestra los tamaños de las instancias de cada grupo. Para cada grupo, la tercera columna contiene los indices de dispersion (Sp), coeficiente de variación (VC) y sesgo (Sk). Las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de cada uno de los índices, en cada uno de los grupos.

Como se puede observar en la Tabla 4.3, para las instancias OPT-I el índice de dispersión promedio varia entre un 50 % y un 60 %, lo cual significa que el porcentaje de ceros en las matrices de costos de las instancias de todos los grupos esta en ese rango. Con respecto a este indicador las instancias no presentan una diferencia estructural clara.

Con respecto al coeficiente de variación promedio se observa que las que tienen un valor mas bajo son las random y las MB seguidas de las special y de las LOLIB y SGB. Con respecto a este indicador las instancias LOLIB y SGB muestran una ligera diferencia estructural con respecto al resto de las instancias.

Con respecto al promedio del valor del sesgo se establece una diferencia similar a la observada en el caso del coeficiente de variación. Se observa que las que tienen un valor mas bajo son las random y las MB seguidas de las special y de las LOLIB y SGB. Con respecto a este indicador las instancias LOLIB y SGB muestran una ligera diferencia estructural con respecto al resto de las instancias.

En el caso de las instancias UB-I (Tabla 4.4), para los primeros cuatro grupos el índice de dispersión promedio es de aproximadamente 50 % y el las instancias special tienen un índice de 74 %. Esto significa que en el caso de los 4 primeros grupos la matriz

Tabla 4.3: Análisis de distribución de instancias OPT-I de LOP

Instancias	Tamaño(s)	Índice	Min	Mediana	Max	Promedio
		Sp.	51.136364	57.924107	85.278926	60.935536
LOLIB	50/60/44/	VC	4.861002	5.70401	17.010597	6.41250606
	56/79	Sk.	10.48144	13.8704705	39.19369	16.3435803
		Sp.	53.368889	54.364444	55.466667	54.3473778
SGB	75	VC	8.316902	8.403378	9.087727	8.4606556
		Sk.	21.296385	21.7391605	22.642443	21.8117239
		Sp.	55.11	56.06	56.62	56.0172
Random AII	100	VC	1.460055	1.478456	1.508897	1.48020904
		Sk.	1.528034	1.598855	1.759874	1.60662028
		Sp.	51.188017	51.497934	52.125	51.5319068
Random B	40/44	VC	1.271145	1.3068875	1.346363	1.307131
		Sk.	0.902559	1.0020255	1.146573	1.00179246
		Sp.	50.6816	51.0318055	53.29	51.3400537
MB	100/150/	VC	1.409664	1.429015	1.492937	1.43692657
	200/250	Sk.	1.272079	1.3255335	1.474479	1.34234897
		Sp.	51.851852	54.656608	80.111111	62.4112144
Special	11-77	VC	1.037749	6.407372	9.133312	4.67452362
		Sk.	0.074125	16.372212	22.675608	10.963135

de costos tiene aproximadamente un 50 % de ceros y en las instancias special se tiene aproximadamente un 75 % de ceros. Con respecto a este indicador las instancias special muestran una ligera diferencia estructural con respecto al resto de las instancias.

Con respecto al coeficiente de variación promedio se observa que las que tienen un valor mas bajo son los tres grupos correspondientes a las instancias random. Las instancias

Tabla 4.4: Análisis de distribución de instancias UB-I de LOP

Instancias	Tamaño(s)	Índice	Min	Mediana	Max	Promedio
Random AI	100/150/200/ 200/500	Sp.	49.86	50.2141665	50.6228	50.2499538
		VC	1.388893	1.40905	1.425383	1.40935779
		Sk.	1.299494	1.337543	1.37876	1.33649832
Random AII	150/200	Sp.	50.333333	52.2054165	54.73	52.3032165
		VC	1.442396	1.466864	1.496384	1.46680452
		Sk.	1.541931	1.621808	1.721194	1.62636756
Random B	50	Sp.	51.24	51.42	51.72	51.434
		VC	1.290915	1.305992	1.322251	1.30697395
		Sk.	0.9314	0.956117	1.02746	0.9582984
XLOLIB	150/250	Sp.	52.28	55.7379555	82.4912	58.5887453
		VC	4.619342	5.5645115	16.447242	6.13970322
		Sk.	10.070898	13.697023	43.692886	16.5069182
Special	31/35/43/ 111/134/163/ 452	Sp.	50.909091	86.940995	98.533558	74.2166164
		VC	1.01835	2.760413	8.556659	2.97795443
		Sk.	0.03637	3.082035	9.805175	2.89175829

special muestran el siguiente nivel mas alto de variación y finalmente las instancias XLOLIB muestran un valor de variación considerablemente alto. Con respecto a este indicador las instancias XLOLIB muestran una clara diferencia estructural con respecto al resto de las instancias.

Con respecto al promedio del valor del sesgo se establece una diferencia similar a la observada en el caso del coeficiente de variación. Los tres primeros grupos muestran un sesgo similar, seguidas de las especiales y las XLOLIB. La distribución de los datos en las instancias XLOLIB es diferente al resto de las instancias.

4.3 Estudio de las estructura de las instancias del Problema de Ordenamiento Lineal con Costos Acumulados

Los resultados del estudio de la estructura de las instancias de LOPCC se muestran en la Tabla 4.5. La primera columna de las tablas contiene el grupo de instancias que corresponde a los índices mostrados. La segunda muestra los tamaños de las instancias de cada grupo. Para cada grupo, la tercera columna contiene los indices de dispersion (Sp), coeficiente de variación (VC) y sesgo (Sk). Las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de cada uno de los índices, en cada uno de los grupos.

Tabla 4.5: Análisis de distribución de instancias LOPCC

Instancias	Tamaño	Índice	Min	Mediana	Max	Promedio
UMTS	16	Sp.	0	0	0	0
		VC	8.201257	15.3826875	15.968719	14.1358752
		Sk.	9.144264	15.902134	15.906097	15.045055
LOLIB	44/50/56/60	Sp.	11.002066	36.1752065	80.630165	37.5747255
		VC	4.101153	4.8471325	16.249833	5.47395102
		Sk.	9.146213	12.7145855	39.156412	15.3375085
Random	35/100/150	Sp.	0.326531	1.004722	1.55102	1.00154381
		VC	0.560426	0.582516	0.597549	0.5815788
		Sk.	-0.084489	-0.000347	0.043129	-0.00494247

Como se puede observar en la Tabla 4.5, el índice de dispersión promedio de las instancias UMTS es cero, el de las instancias LOLIB es 37.5 y el de las instancias Random es de 1.0. En este caso se observa una gran diferencia en la estructura de las diferentes instancias ya que las matrices de costos de las instancias LOLIB contienen aproximadamente un 40 % de ceros, a diferencia de las random que contienen un 1 % de ceros y las UMTS que no contienen ceros.

Con respecto al coeficiente de variación promedio se observa que las instancias UMTS muestran el valor mas alto sguidas de las instancias LOLIB y las instancias Random. Con respecto a este indicador las instancias UMTS muestran una clara diferencia con respecto al resto de las instancias.

Con respecto al promedio del valor del sesgo se observa que las instancias UMTS y LOLIB muestran un valor similar, mucho mayor que el observado para las instancias Random. Con respecto a este indicador las instancias UMTS y LOLIB muestran una clara diferencia estructural con respecto a las instancias Random.

CAPÍTULO 5

Estudio de la superficie de búsqueda

En este capítulo se presenta el estudio de la superficie de búsqueda de las instancias de los problemas LOP y LOPCC. Se expone la forma en que se llevó a cabo, las características del equipo de experimentación, así como algoritmos usados en estas evaluaciones. Como punto final en este capitulo se presenta un grupo de comentarios en forma de conclusión.

5.1 Descripción del estudio de la longitud de la correlación de superficie para LOP y LOPCC

En este estudio se analiza la rugosidad de las instancias de LOP y LOPCC, y se proponen técnicas apropiadas para resolverlas.

Para realizar las pruebas se utilizó un servidor con un procesador Intel XEON 3.06 GHz, 4GB de RAM y 70 GB de disco duro. El ambiente de programación usado fue Visual Studio 2005 y se usó el compilador Visual C++.

Para este estudio se implementó un algoritmo que realiza una caminata aleatoria de un millón de pasos sobre la superficie de soluciones. Posteriormente, se analizó la superficie de búsqueda de las instancias de acuerdo a la trayectoria seguida por tres diferentes algoritmos: búsqueda local iterada (ILS), Búsqueda Tabú y un algoritmo Memético. La caminata para el análisis en estos últimos algoritmos fue limitada a un tiempo de 60 segundos y un determinado número de iteraciones sin mejora. Para los resultados de estas pruebas se obtuvo el valor objetivo mínimo, mediana, máximo y el promedio de cada grupo de instancias.

5.1.1 Longitud de correlación de superficie con una caminata aleatoria

Para esta prueba se implementó un algoritmo aleatorio. Este algoritmo recorre la superficie de soluciones paso a paso, en donde cada paso corresponde a tomar un elemento i cualesquiera de la permutación actual y moverlo a una posición aleatoria j, tal que $i \neq j$. Este algoritmo se puede observar en la Figura 5.1

La caminata comienza con una solución aleatoria y su correspondiente valor de la función objetivo. Posteriormente se realizan m movimientos sobre la superficie, cada movimiento es un paso en la caminata. Una vez hecho un movimiento, se lleva a cabo el cálculo del valor de la función objetivo el cual es usado para el análisis de la caminata. La longitud de la caminata aleatoria realizada fue de un millón de pasos. Esto generó una serie con los valores objetivo de un millón de soluciones aleatorias, la cual es utilizada para calcular la longitud de la correlación de la superficie de búsqueda recorrida

π = *SolucionInicialAleatoria()*

$FO[0]$ =*CalculoFO(π)*

for *i=1...(m)* **do**(donde m es la cantidad de movimientos aleatorios de inserción a realizar(pasos))

{

π =*MovimientoInsercionAleatorio(π)*

$FO[i]$ =*CalculoFO(π)*

}

end for

Figura 5.1: Estructura básica del algoritmo Aleatorio aplicado a LOP y LOPCC

5.1.2 Longitud de la correlación de la superficie para LOPCC con un algoritmo de Búsqueda Local Iterada (ILS)

Para complementar el estudio de correlación de superficie de soluciones, se implementó un algoritmo de Búsqueda Local Iterada ILS. Para este algoritmo se analizó el conjunto de soluciones que conforman su trayectoria de búsqueda. En la Figura 5.2 se presenta el algoritmo básico ILS y sus características.

Para la solución inicial se utiliza una construcción voráz, mediante la cual se construyen 10 soluciones y se selecciona la mejor. Posteriormente a la solución seleccionada se le aplica una búsqueda local que utiliza la vecindad N_2 [5]. Posteriormente se inicia un ciclo en el que se perturba la solución actual y a la solución perturbada se le aplica búsqueda local. Este ciclo continua hasta que se cumple la condición de tiempo o de ciclos sin mejora. Durante el proceso los valores objetivo de las soluciones visitadas son almacenados para formar una serie de valores, la cual es utilizada para calcular la

π = *ConstrucciónInicial()*

π = *BúsquedaLocal_ $N_2(\pi)$*

While (condición de paro)

{

π' = *Perturbación(π)*

π' = *BúsquedaLocal_ $N_2(\pi')$*

π' = *CriterioDeAceptación(π')*

}

Figura 5.2: Estructura básica del algoritmo ILS LOPCC

longitud de la correlación de la superficie de búsqueda. En este caso la longitud de la serie esta determinada por las condiciones de parada del algoritmo.

- Perturbación. Permite realizar 10 movimientos de inserción para tratar de encontrar una solución que mejore a la actual.

- Búsqueda local. Se le aplica una búsqueda local, utilizando la vecindad N_2, a la mejor solución encontrada hasta el momento.

- Criterio de aceptación. Si la solución obtenida en la búsqueda local tiene una mejora significativa en comparación con la mejor solución obtenida hasta el momento (hasta antes de esta búsqueda), se parte de está para la nueva perturbación, en caso contrario se continúa perturbando la solución que se tenía antes de realizar la búsqueda local.

5.1.3 Longitud de la correlación de la superficie para LOPCC con un algoritmo de Búsqueda Tabú

Para este análisis se implementó el algoritmo de Búsqueda Tabú reportado por Duarte en [4]. La Figura 5.3 muestra el pseudocódigo de dicho algoritmo.

ConstructSolInicial()
Actualizo la mejor solución encontrada(p_overall_best)
While*(!=CantidadIteracionesSinMejora)*
{
  Intensificación(p_overall_best)
  Se verifica si se encontró una mejor solución global y en caso de ser así se reinicia el iterador global
  Diversificación(p_overall_best)
  Se verifica si se encontró una mejor solución y en caso de ser así se reinicia el iterador global
}

Figura 5.3: Estructura básica del algoritmo de búsqueda Tabú LOPCC

El algoritmo inicia aplicando una heurística para generar cien soluciones candidatas de entre las que se selecciona la mejor como solución inicial para el proceso de Búsqueda Tabú. Posteriormente se inicia un proceso de intensificación en el cual la solución actual es mejorada aplicando repetidamente una búsqueda local hasta que la solución no mejore durante un cierto número de iteraciones. Durante el proceso de intensificación se registran las cuatro mejores soluciones visitadas, a las que se les denomina soluciones elite. Posteriormente se determina la probabilidad de que cada sector ocupe una determinada posición en las soluciones elite. Esta distribución de probabilidad es utilizada para generar una nueva solución en la cual se selecciona para cada posición el sector menos probable, tratando de generar una solución diferente a las soluciones elite. Este proceso de diversificación conjuntamente con el de intensificación se realizan repetidamente hasta que la solución global no mejore en un cierto número de iteraciones o se cumpla la condición de tiempo. Durante el proceso los valores objetivo de las soluciones visitadas son almacenados para formar una serie de valores, la cual es utilizada para calcular la longitud de la correlación de la superficie de búsqueda. En este caso la longitud de la serie esta determinada por las condiciones de parada del algoritmo.

5.1.4 Longitud de la correlación de la superficie superficie para LOPCC con un algoritmo Memético

Para este análisis se implementó el algoritmo Memético reportado por Schiavinotto en [2]. La Figura 5.4 muestra el pseudocódigo de dicho algoritmo.

El algoritmo inicia generando aleatoriamente una población inicial a cuyos individuos se les aplica búsqueda local. Durante un cierto numero de generaciones se realiza la selección y cruzamiento de individuos de la población para generar un conjunto de nuevos individuos a los que se aplica búsqueda local. Los mejores individuos resultados del cruzamiento son incorporados a las población y se determina el valor promedio de los individuos en esta. Si durante un cierto número de generaciones este promedio no cambia se realiza una regeneración de la población para diversificarla. En este proceso mantiene en la población únicamente el mejor individuo y se completa la población con individuos generados aleatoriamente. A todos los individuos de la población generada se les aplica una búsqueda local. El proceso de cruzamiento, actualización y diversificación de la población continua hasta que se agote el limite de 60 segundos. Durante el proceso de búsqueda los valores objetivo de los individuos generados son almacenados para formar una serie de valores, la cual es utilizada para calcular la longitud de la correlación de la superficie de búsqueda.

Población $\leftarrow$ *{ }*
for *i=1...m* **do** (donde m es el número de individuos)
  $\pi \leftarrow$ *BúsquedaLocal(GeneraBúsquedaLocal())*
  Población $\leftarrow$*Población* $\cup\{\pi\}$
end for
repeat
  NuevaPoblación $\leftarrow$ *{ }*
  for $\leftarrow 1 \ldots$ *Cruzamientos* **do**
    Tomar π_a, π_b *de la población*
    NuevaPoblación $\leftarrow$ *NuevaPoblación* $\cup\{$BúsquedaLocal(Cruzamiento(π_a, π_b))$\}$
  end for
  Población $\leftarrow$*SeleccionarMejor(Población* $\cup$*NuevaPoblación, m)*
  if mismo promedio de solución en un tiempo determinado **then**
  (Diversificación)
    Población $\leftarrow$*SeleccionarMejor(Población* $\cup$*NuevaPoblación, m)*
    for *i=1...(m-1)* **do**
      $\pi \leftarrow$ *BúsquedaLocal(GeneraBúsquedaLocal())*
      Población $\leftarrow$*Población* $\cup\pi$
    end for
  end if
until condición de parada

Figura 5.4: Estructura básica del algoritmo Memético LOPCC

5.2 Estudio de la longitud de la correlación de la superficie para LOP

Los resultados del estudio de longitud de la correlación de la superficie de las instancias OPT-I de LOP se presentan en la Tabla 5.1, y de las instancias UB-I de LOP en la Tabla 5.2. La primera columna de las tablas muestra el grupo de instancias evaluadas y las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de la longitud de la correlación de superficie de cada uno de los grupos.

Tabla 5.1: Estadísticas (mínimo, mediana, máximo y promedio) de la longitud de correlación de la superficie recorrida con una caminata aleatoria para las instancias de LOP con óptimo conocido (OPT-I), normalizada con respecto al tamaño de las instancias (l/n)

Instancias	Min	Mediana	Max	Promedio
LOLIB	0.754377932	0.801170305	0.845766017	0.803625329
SGB	0.468937467	0.480153693	0.487862187	0.480184286
RandomAII	0.92745571	0.96389696	0.99162036	0.962390648
RandomB	0.662834409	0.764576163	0.859907364	0.763226448
MB	0.92426443	0.961377345	1.01078716	0.962466214
Special	0.368881091	0.7472063	0.855090145	0.718311296

Como se observa para las OPT-I, las instancias MB y RandomAII tienen la menor rugosidad, las LOLIB, RandomB y Special muestran un nivel intermedio y las instancias SGB muestran el mayor grado de rugosidad. Para el grupo de instancias UB-I las RandomAII presentan un índice muy bajo de rugosidad, seguidas de las instancias Special, RandomAI, RandomB y XLOLIB, las cuales muestran un nivel intermedio de rugosidad.

Tabla 5.2: Estadísticas (mínimo, mediana, máximo y promedio) de la longitud de correlación de la superficie recorrida con una caminata aleatoria para las instancias de LOP con óptimo no conocido (UB-I), normalizada con respecto al tamaño de las instancias ($1/n$)

Instancias	Min	Mediana	Max	Promedio
Random AI	0.68349987	0.735511563	0.80200816	0.736658123
RandomAII	0.943432387	0.97562477	1.00206974	0.974528778
RandomB	0.68407232	0.7331988	0.8141086	0.733814899
XLOLIB	0.694615927	0.74228617	0.769354673	0.738359149
Special	0.452687613	0.852103296	0.904545478	0.750952142

5.3 Estudio de la longitud de correlación de la superficie para LOPCC

5.3.1 Estudio de la longitud de correlación de la superficie con un algoritmo aleatorio

Los resultados del estudio de la longitud de la correlación de la superficie de las instancias de LOPCC se muestran en la Tabla 5.3. La primera columna de las tablas muestra el grupo de instancias evaluadas y las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de la longitud de la correlación de la superficie de cada uno de los grupos.

De acuerdo a los resultados obtenidos en el análisis, se observa que las instancias LOLIB son las más rugosas con un índice promedio de 0.138567. Después de éstas se encuentran las instancias UMTS con un índice de rugosidad promedio de 0.419291, y finalmente están las instancias Random con el índice de rugosidad más bajo de 0.460567.

Tabla 5.3: Estadísticas (mínimo, mediana, máximo y promedio) de la longitud de correlación de la superficie recorrida con un algoritmo aleatorio para las instancias de LOPCC, normalizada con respecto al tamaño de las instancias (l/n)

Instancias	Min	Mediana	Max	Promedio
UMTS	0.202872	0.421042073	0.701806438	0.419291625
LOLIB	0.045354	0.116509	17.938909	0.138567486
Random	0.34012036	0.438208715	0.679978286	0.460567816

5.3.2 Estudio de la longitud de la correlación de la superficie con un algoritmo ILS

Los resultados del estudio de la longitud de la correlación de la superficie con un algoritmo ILS se muestran en la Tabla 5.4. La primera columna de la tabla muestra el grupo de instancias evaluadas y las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de la longitud de la correlación de la superficie de cada uno de los grupos.

Tabla 5.4: Estadísticas (mínimo, mediana, máximo y promedio) de la longitud de correlación de la superficie recorrida con un algoritmo ILS para las instancias de LOPCC, normalizada con respecto al tamaño de las instancias (l/n)

Instancias	Min	Mediana	Max	Promedio
UMTS	0.025462813	0.319324781	1.153544125	0.399843863
LOLIB	0.010412614	0.1690425	1.096410286	0.283325772
Random	0.025362887	0.380876657	1.3910062	0.407177699

De acuerdo a los resultados del análisis aplicado al algoritmo ILS, éste percibe menos rugosas las instancias LOLIB que la caminata aleatoria, mientras que en las demás instancias el efecto es menor. Siendo aún las instancias Lolib las mas rugosas y las instancias Random las menos rugosas.

5.3.3 Estudio de la longitud de la correlación de la superficie con un algoritmo de búsqueda Tabú

Los resultados del estudio de la longitud de la correlación de la superficie con un algoritmo de búsqueda Tabú se muestran en la Tabla 5.5. La primera columna de la tabla muestra el grupo de instancias evaluadas y las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de la longitud de la correlación de la superficie de cada uno de los grupos.

Tabla 5.5: Estadísticas (mínimo, mediana, máximo y promedio) de la longitud de correlación de la superficie recorrida con un algoritmo de búsqueda Tabú para las instancias de LOPCC, normalizada con respecto al tamaño de las instancias (l/n)

Instancias	Min	Mediana	Max	Promedio
UMTS	0.061571	0.228395	0.735723	0.25980694
LOLIB	0.012063	0.0639595	1.166656	0.209886354
Random	0.028086	0.181764	0.799034	0.23711568

Para el algoritmo de Búsqueda Tabú la rugosidad de las instancias LOLIB es menor que la rugosidad obtenida de la caminata aleatoria, y que en las instancias menos rugosas (UMTS y Random) el índice de rugosidad subió, a tal grado que la rugosidad de los tres tipos de instancias son muy similares. Para este algoritmo las instancias más rugosas son las Lolib, seguidas de cerca por las instancias Random y UMTS respectivamente.

5.3.4 Estudio de la longitud de la correlación de la superficie con un algoritmo Memético

Los resultados del estudio de la longitud de la correlación de la superficie con un algoritmo Memético se muestran en la Tabla 5.6. La primera columna de la tabla muestra el grupo de instancias evaluadas y las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de la longitud de la correlación.

Tabla 5.6: Estadísticas (mínimo, mediana, máximo y promedio) de la longitud de correlación de la superficie recorrida con un algoritmo Memético para las instancias de LOPCC, normalizada con respecto al tamaño de las instancias ($1/n$)

Instancias	Min	Mediana	Max	Promedio
UMTS	0.028914	0.1515745	0.342634	0.16378875
LOLIB	0.001652	0.033361	0.138172	0.043238612
Random	0.011269	0.074741	0.48672	0.141495267

De acuerdo a los resultados el algoritmo Memético percibe con mayor rugosidad que el algoritmo aleatorio, a todas las instancias. Para este algoritmo las instancias UMTS son las menos rugosas, y las instancias LOLIB, las más rugosas.

5.3.5 Análisis de la cantidad de soluciones visitadas por cada algoritmo

En este estudio se analiza la capacidad de búsqueda de los métodos usados anteriormente en el análisis de rugosidad. Este análisis se basa en observar como se desempeñan diferentes algoritmos tomando en cuenta tipo de rugosidad que presenta la superficie de soluciones. Las características de los algoritmos utilizados son las siguientes: el algoritmo de Búsqueda Local Iterada tiene la característica de ser un algoritmo de búsqueda con un alto grado de intensificación, el algoritmo Búsqueda Tabu realiza de una forma balanceada tanto intensificación como diversificación, y el algoritmo Memético realiza un alto grado de diversificación. Las Tablas 5.7, 5.8 y 5.9 muestran la cantidad de soluciones visitadas por cada algoritmo de acuerdo al grupo de instancias. La primera columna de las tablas muestra el grupo de instancias evaluadas y las siguientes cuatro columnas contienen el valor mínimo, la mediana, el máximo y el promedio de la cantidad de soluciones visitadas por el algoritmo.

Tabla 5.7: Cantidad de soluciones visitadas por el algoritmo ILS

Instancias	Min	Mediana	Max	Promedio
UMTS	482	562	802	565
LOLIB	1724	2560	6298	2922
Random	1620	5020	12190	5460

Tabla 5.8: Cantidad de soluciones visitadas por el algoritmo de Búsqueda Tabú

Instancias	Min	Mediana	Max	Promedio
UMTS	839	1359.5	6260	1627.18
LOLIB	2983	5401	17067	6115.40816
Random	2757	5046	20731	8617.54667

Tabla 5.9: Cantidad de soluciones visitadas por el algorítmo Memético

Instancias	Min	Mediana	Max	Promedio
UMTS	7444	8356	24692	10585.26
LOLIB	21167	50583	139574	54788.8571
Random	19315	38536	70736	37887.8267

La Tabla 5.10 muestra una comparativa de el espacio que cubre cada uno de los algoritmos utilizados en función de la cantidad de soluciones visitadas. La primera columna de la tabla muestran los grupos de instancias. La segunda muestra los tamaños de las instancias de cada grupo. La tercera muestra la valoración de rugosidad (menos-mas). La cuarta, quinta y sexta muestran la cantidad de soluciones visitadas (de acuerdo a la media) por cada algoritmo de acuerdo al tipo de instancia.

Tabla 5.10: Número de soluciones exploradas por los diferentes algoritmos utilizados, y su relación con el tipo de la superficie de búsqueda asociada (más o menos rugosa) a cada grupo de instancias.

Instancias	Tamaños	Rugosidad	ILS	Tabú	Memético
UMTS	16	menos	565	1627	10585
Random	35/100/150	menos	**5460**	bf 8617	**37887**
LOLIB	44/50/56/60	mas	**2922**	bf 6115	**54788**

Se observa que el algoritmo ILS visita una mayor cantidad de soluciones en las instancias Random que en las LOLIB. Por el contrario el algoritmo Memético visita una mayor cantidad de soluciones en las instancias Lolib que en las Random. Por lo que pareciera ser que los algoritmos intensificadores son una mejor opción para resolver instancias cuya superficie de búsqueda es poco rugosa, y por el contrario los algoritmos diversificadores son más apropiados para instancias cuya superficie es más rugosa.

La evidencia experimental recabada muestra que un algoritmo diversificador logra un mejor desempeño sobre las instancias de mayor rugosidad. En las instancias de menor rugosidad se desempeñan mejor los algoritmos intensificadores. Esto sugiere una metodología para la selección del balance de intensificación y diversificación en un algoritmo. La Tabla 5.11 muestra la relación entre la rugosidad de superficie de búsqueda asociada a las instancias y las características más adecuadas del algoritmo de solución.

Una metodología tal debe incluir al menos los siguientes pasos:

1. Agrupar las instancias para integrar grupos de tamaño similar.

Tabla 5.11: Características de un algoritmo en base a la rugosidad de las instancias

Instancias	Rugosidad	Característica
I1	menos	mayor intensificación
I2	más	mayor diversificación

2. Utilizando la longitud de correlación normalizada, determinar la rugosidad relativa de las superficies de búsqueda asociada a cada grupo de instancias.

3. Identificar las instancias más y menos rugosas.

4. Para las instancias menos rugosas utilizar algoritmos intensificadores, y para las instancias más rugosas, algoritmos diversificadores.

CAPÍTULO 6

Distancia de aptitud

6.1 Descripción del estudio de distancia de aptitud

En este estudio se analizó la ubicación relativa de los óptimos locales en relación a las mejores soluciones conocidas. Para las búsquedas locales utilizó la vecindad N_2 [5].

El estudio de distancia de aptitud permite evaluar la calidad de un proceso de búsqueda local aplicada a la solución de un conjunto de instancias de un problema. Refleja qué tan buena es la búsqueda local, mostrando la distribución y ubicación relativa de óptimos locales en relación al mejor valor conocido o al óptimo global.

Para realizar el estudio, con una instancia dada, se generan un número determinado soluciones candidatas aleatorias, a cada una de las cuales se les aplica una búsqueda local. Este proceso dirige la búsqueda de una solución candidata al óptimo local más cercano. La cantidad de búsquedas locales fue determinada de acuerdo al tamaño de cada instancia, ejecutando 13,000 búsquedas para las instancias pequeñas (UMTS, LOLIB y Random de tamaño 35) y 1,000 búsquedas para las instancias de mayor tamaño (Random de tamaños 100 y 150).

Para realizar las pruebas se utilizó un servidor con un procesador Intel XEON 3.06 GHz, 4GB de RAM y 70 GB de disco duro. El ambiente de programación usado fue Visual Studio 2005 y se usó el compilador Visual C++.

La Figura 6.1 muestra el algoritmo general usado para la obtención de óptimos locales.

s=0
While(*s<CantSols*)
{
  π = *SoluciónAleatoria()*
  π' = *BúsquedaLocal*(π)
  $s++$
}

Figura 6.1: Algoritmo para la obtención de óptimos locales

6.2 Estudio de la distancia de aptitud

En el proceso para calcular la distancia de aptitud de las diferentes instancias se puede obtener información adicional tal como: óptimos locales distintos, mejores soluciones conocidas encontradas, etc. Previo a la presentación de los resultados relativos a la distancia de aptitud, se muestran los resultados observados para diferentes indicadores. El propósito de generar esta información previa es tratar de encontrar una relación con las características observadas en el estudio de la estructura de las instancias y la superficie de búsqueda. En todos los casos los resultados muestran para cada grupo de instancias, los valores mínimo, mediana, máximo y el promedio del indicador correspondiente.

La Tabla 6.1 muestra la cantidad de óptimos locales distintos encontrados al aplicar las búsquedas locales.

Tabla 6.1: Cantidad de óptimos locales distintos encontrados

Caraceteristica	Min	Mediana	Max	Promedio
UMTS	1	3	75	7.07
LOLIB	1	103.5	6282	486.57
Random (35)	8	60	522	106.56

De acuerdo a los resultados obtenidos se observa que la cantidad de óptimos locales distintos es muy pequeño para las instancias LOLIB con un promedio de 7, mientras que para las instancias Random y LOLIB es de 106.56 y 486.57 respectivamente.

En este análisis se puede observar la relación entre rugosidad y óptimos locales distintos encontrados. Una superficie con mayor rugosidad tiene un mayor número de óptimos locales distintos. En este caso se observa que las instancias LOLIB tienen el mayor número de óptimos locales distintos y son las de mayor rugosidad.

La Tabla 6.2 muestra el número de búsquedas locales que llegan al mejor conocido.

Tabla 6.2: Número de búsquedas locales que llegan al mejor conocido

Caraceterística	Min	Mediana	Max	Promedio
UMTS	1268	6949	13 000	7 318.37
LOLIB	16	2021.5	13 000	3002.68421
Random (35)	147	1013	4280	1276.32

Como se observa en la tabla las instancias UMTS logran llegar en más del 50 % al mejor conocido. Para las instancias LOLIB y Random, las búsquedas locales encuentran óptimos locales en aproximadamente un 25 % y un 10 % de las búsquedas realizadas. De acuerdo a este indicador las instancias Random muestran una mayor dificultad para alcanzar los óptimos locales.

La Tabla 6.3 muestra el número de búsquedas locales que llegan a la mejor solución conocida con diferentes permutaciones.

Tabla 6.3: Número de búsquedas locales que llegan al mejor conocido con diferentes permutaciones

Caraceterística	Min	Mediana	Max	Promedio
UMTS	1	1	1	1
LOLIB	16	896	12836	2270.1578
Random (35)	1	10	261	31.88

En la tabla se observa que las instancias LOLIB presentan el índice más elevado con un promedio de 2270.1578. Para las instancias Random este indicador tiene un promedio de 31.88. Para las instancias UMTS se llega a la mejor solución conocida con una única permutación.

Este análisis puede ser relacionado con el análisis de dispersión, ya que al tener un índice elevado de ceros repetidos en la instancia, es posible llegar a la misma solución intercambiando posiciones ocupadas por ceros. Por lo que, al tener las instancias LOLIB casi un 50 % de ceros en su matriz, se pueden generar diferentes soluciones con mismo valor objetivo.

Por ultimo la Tabla 6.4 muestra la *distancia de aptitud* minima, mediana, máxima y promedio para los diferentes grupos de instancias considerados.

Tabla 6.4: Distancia de aptitud

Caraceteristica	Min	Mediana	Max	Promedio
UMTS	0.36851	0.9043	1	0.845346
LOLIB	0.19149	0.60682	0.99262	0.5929935
Random (35)	0.32286	0.9423	0.64938	0.640928

De acuerdo a la prueba de distancia de aptitud realizada, se observa que la calidad de las soluciones obtenidas en las instancias UMTS es la más elevada. Como se señaló previamente las instancias LOLIB alcanzan un número mayor de óptimos locales que las Random. Sin embargo, como se observa en la tabla las soluciones de las instancias LOLIB son de menor calidad que las soluciones de las instancias Random. Esto se explica también como consecuencia del alto porcentaje de ceros de las instancias LOLIB, ya que el proceso de búsqueda se puede estancar en soluciones de baja calidad cuando se realizan una serie de movimientos que solo involucran las posiciones que contienen ceros.

La Tabla 6.5 presenta por separado los resultados del análisis realizado para las instancias más grandes (Random 100 y Random 150). Estos análisis son: (1) número de óptimos locales distintos encontrados,(2) número de búsquedas locales que llegan al mejor conocido, (3) número de óptimos locales que llegan al mejor conocido con diferentes permutaciones, y (4) el índice de distancia de aptitud.

Tabla 6.5: Características del estudio de Distancia de aptitud para las instancias Random 100 y 150

Indicador	Min	Mediana	Max	Promedio
1	840	1000	1000	999.87230
2	0	0	0	0
3	0	0	0	0
4	-0.001940	0.431991	0.7694	0.412063

Por último en el análisis realizado a las instancias Random de longitud 100 y 150, se observa que casi en el 100 % de sus búsquedas encuentra soluciones distintas sin llegar al mejor conocido u óptimo global. Finalmente se observa que la calidad que presentan estas búsquedas locales es más baja que el obtenido en las instancias más pequeñas.

CAPÍTULO 7

Conclusiones y trabajos futuros

En este capítulo se describen las aportaciones más relevantes de este proyecto de investigación así como los trabajos futuros identificados en el proceso.

7.1 Principales aportaciones

1. En el contexto del Problema de Ordenamiento Lineal, se aporta una actualización del estudio de la estructura y de la superficie de búsqueda de las instancias del Problema de Ordenamiento Lineal realizado por Schiavinotto [2]. Dicho trabajo no incluye todas las instancias consideradas en el marco de referencia reportado por Martí en [3]. En esta tesis se presenta un estudio actualizado de la estructura de las instancias y de la longitud de correlación de la superficie de búsqueda del Problema de Ordenamiento Lineal (sección 4.2 y 5.2).

2. En el contexto del Problema de Ordenamiento Lineal con Costos Acumulados, se aportan los primeros estudios de la estructura y de superficie de búsqueda de las instancias del problema. Como resultado de estos estudios se caracterizan las necesidades de intensificación y diversificación de los algoritmos de solución y se

propone una metodología para determinar los requerimientos de intensificación y diversificación de los algoritmos de solución de las instancias del problema (secciones 5.3.1 - 5.3.5).

3. En el contexto del análisis de la longitud de la correlación de la superficie de búsqueda, como parte del estudio correspondiente para el Problema de Ordenamiento Lineal con Costos Acumulados, se observa que la relación relativa entre la rugosidad de los diferentes tipos de instancias, no parece depender del algoritmo que se utilice para la generación de las soluciones candidatas. En este trabajo se muestra, con respecto a las instancias más grandes, que todos los algoritmos utilizados (aleatorio, ILS, Búsqueda Tabú y Memético) perciben a las instancias LOLIB como las más rugosas y a las Random como las menos rugosas. La evidencia experimental recabada en este estudio muestra que la geometría de la superficie (al menos para las instancias más grandes) no depende del algoritmo que se esté utilizando (sección 5.3).

7.2 Trabajos futuros

Algunos de los trabajos con los que se puede continuar esta investigación son los siguientes:

1. Actualizar el estudio de distancia de aptitud para el problema de LOP.
2. Realizar un estudio de la relación del algoritmo de solución de un problema con la geometría de la superficie de búsqueda que incluya diversos problemas y algoritmos.
3. Diseñar metaheurísticas que adapten sus niveles de intensificación y diversificación, en función de la geometría de la superficie de búsqueda.

4. Para LOPCC, generar instancias de mayor tamaño para analizarlas.

5. Para LOPCC, realizar el estudio de distancia de aptitud utilizando diferentes métricas de distancia entre permutaciones.

Referencias

[1] Bertacco, L., Brunetta, L., Fischetti, M.: The linear ordering problem with cumulative costs. Eur. J. Oper. Res. **189(3)** (2008) 1345–1357

[2] Schiavinotto, T., Stützle, T.: The linear ordering problem: Instances, search space analysis and algorithms. Journal of Mathematical Modelling and Algorithms **Volume 3** (2005) pp. 367–402(36)

[3] Martí, R., Reinelt, G., Duarte, A.: A benchmark library and a comparison of heuristic methods for the linear ordering problem. Biological Cybernetics (2009) 27

[4] Duarte, A., Laguna, M., Martí, R.: Tabu search for the linear ordering problem with cumulative costs. Springer Science+Business Media (2009)

[5] Laguna, M., Martí, R., Campos, V.: Intensification and diversification with elite tabu search solutions for the linear ordering problem. (1998)

[6] Reinelt, G.: The linear ordering problem: algorithms and applications. Research and exposition in mathematics 9, Heldermann **Volume 9** (1985)

[7] Campos, V., Glover, F., Laguna, M., Martí, R.: An experimental evaluation of a scatter search for the linear ordering problem. (1999)

[8] Quesada, I.M.B., Vergara, S.J.C: Estadistíca básica con aplicaciones en Ms Excel. (2007)

[9] Kim, T., White, H.: On more robust estimation of skewness and kurtosis: Simulation and application to the s&p500 index. (2003)

[10] Stützle, T., Hoos, H.: Max-min ant system. Future Generation Computer Systems (2000) 16(8):889–914

[11] Boese, K.: Models for iterative global optimization. (1996)

[12] Schiavinotto, T., Stützle, T.: A review of metrics on permutations for search landscape analysis. Computers and Operations Research **Volume 34** (2007) 3143–3153 (10)

[13] Reeves, C.: Landscapes, operators and heuristic search. Annals of Operational Research (1999) 86:473–490

[14] Weinberger, E.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics (1990) 325–336

[15] Stadler, P.: Towards a theory of landscapes. Complex Systems and Binary Networks **461** (1995) 77–163

[16] Angel E., Zissimopoulos, V.: On the classification of np-complete problems in terms of their correlation coefficient. (2000) 99:261–277

[17] Duarte, M.A., Pantrigo, F.J.J., Gallego M.C.: Metaheurísticas. Servicio de Publicaciones de la Universidad Rey Juan Carlos – Dykinson, S.L. (2007)

[18] Righini, G.: A branch-and-bound algorithm for the linear ordering problem with cumulative costs. (2006)