

# **EXPERIENCIA COMPUTACIONAL CON UN ALGORITMO DE LOCALIZACIÓN-ASIGNACIÓN PARA UN PROBLEMA DE OPTIMIZACIÓN DE SISTEMAS TERRITORIALES**

**Andrés Castrillón Escobar, Roger Z. Ríos-Mercado<sup>1</sup>, Ada M. Álvarez-Socarrás<sup>1</sup>**

<sup>1</sup>Programa de Posgrado en Ingeniería de Sistemas, Universidad Autónoma de Nuevo León,  
San Nicolás de los Garza, N. L., México  
{Andrés, Roger, Adita}@yalma.fime.uanl.mx

## **Resumen**

El propósito de este artículo es el de analizar computacionalmente un método para la resolución de un problema de diseño de territorios en el mundo real para una empresa de distribución de bebidas. El problema consiste en encontrar una partición de un conjunto de unidades geográficas en un determinado número de territorios de tal manera que se minimice una medida de dispersión. Además, los territorios que se formen deben de satisfacer un criterio de balanceo con respecto a dos diferentes medidas, esto es, se busca que los territorios tengan la misma cantidad de clientes y la misma cantidad de productos a entregar, permitiendo cierta tolerancia. Una metodología heurística basada en una técnica de localización-asignación fue desarrollada para resolver este problema. En los experimentos computacionales hemos encontrado que esta heurística otorga soluciones de calidad, incluso en los casos de gran tamaño.

**Palabras clave:** Heurística, localización-asignación, diseño territorial

## **Abstract**

The problem addressed in this paper is to computationally analyze a method for solving a territorial design problem of the real world in a beverage distribution firm. The objective is to find a partition of a set of geographical units into a given number of territories in such a way that minimizes a measure of dispersion. In addition, the territories must meet a criterion of balancing on two different measures, the territories need to have the same number of customers and the same number of deliverables, having this two measures a tolerance. A heuristic methodology based on a location-allocation technique was developed to solve this problem. In the empirical work we found that this heuristic gives good quality solutions, even in large instances.

**Keywords:** Heuristic, location-allocation, territorial design

## **1 Introducción**

El diseño de territorios consiste básicamente en la agrupación de áreas geográficas pequeñas en grandes grupos geográficos llamados territorios de acuerdo a ciertos criterios de planificación.

El problema estudiado en este trabajo fue motivado por una aplicación del mundo real de una empresa de distribución de bebidas embotelladas. Dado un conjunto de manzanas o unidades básicas con conocidos atributos medibles, tales como el número de clientes y la demanda del producto, la empresa tiene que dividir geográficamente a sus clientes en un número fijo de  $p$  territorios en la ciudad de acuerdo con ciertos criterios de planificación. Uno de estos criterios es el de equilibrar el territorio, es decir, la empresa desea obtener territorios equilibrados (similar en tamaño) con respecto a cada una de las dos medidas de actividad diferente (número de clientes y la demanda del producto). La conectividad del territorio también es necesaria, es decir, para cada par de unidades básicas en el territorio, tiene que haber un camino totalmente contenido en ese territorio. Además, se desean territorios compactos, es decir, los clientes dentro de un territorio deben estar lo más cerca posible entre sí. Este criterio se logra en general, minimizando una función de dispersión.

En este trabajo se lleva a cabo una evaluación computacional de un algoritmo diseñado para resolver el problema en cuestión, el cual fue originalmente propuesto por Segura-Ramiro [1]. En la sección 2 se expondrá la importancia y repercusiones que tiene este problema y varios estudios que han hecho diferentes autores a trabajos relacionados con el mismo. En la sección 3 se expondrán cuáles son las metas a alcanzar con este artículo, mientras en las secciones 4 y 5 se

describen, respectivamente la metodología que se usó para abordar este problema y los resultados computacionales que se obtuvieron al aplicar la metodología.

## **2 Justificación**

Este tipo de problemas pertenece a la familia de los problemas de distritos que tienen muchas aplicaciones diferentes, tales como distritos políticos, el diseño de territorios escolares, servicios sociales, servicios de emergencia, las ventas y el diseño del área de servicio. Un extenso estudio reciente de los problemas de distritación y metodologías para encontrar buenas soluciones a estos problemas se puede encontrar en Kalcsics, Nickel y Schröder [2].

## **3 Objetivo del problema**

El problema abordado tiene por objetivo encontrar una partición de un conjunto de unidades geográficas en un determinado número de territorios, de tal manera de reducir al mínimo una medida de dispersión de los territorios formados. Además, los territorios deben ser equilibrados con respecto a dos medidas diferentes. En la Tesis de Segura-Ramiro [1] se aborda con mayor detalle el modelo matemático así como la relajación del modelo.

## **4. Metodología**

La heurística propuesta se basa en un procedimiento de localización-asignación propuesto por Segura Ramiro [1]. Este consta de tres fases iterativas: en primer lugar en la fase de ubicación, los centros de los territorios son elegidos, y luego en una fase de asignación las áreas básicas son asignadas a estos centros. Estas dos fases se combinan con una fase de búsqueda local con el fin de mejorar la calidad de la solución que se encuentra en la fase de asignación. Las tres fases se realizan iterativamente hasta que transcurra un

número determinado de iteraciones sin mejorar la solución. En la Figura 1 se muestra a groso modo el algoritmo compuesto por tres fases las cuales se describen con mayor profundidad en las siguientes secciones

---

```

INPUT Set of initial centers:  $V_c$ .
OUTPUT: Best found solution  $S_{best}$ .
1:  $S_{best} = \emptyset, S_{act} = \emptyset, I = 0$ 
2: WHILE  $I \leq m$  DO
3:    $V_c \leftarrow \text{LOCATION } (S_{act})$ ;
4:    $S_{act} \leftarrow \text{ALLOCATION } (V_c)$ ;
5:    $S_{act} \leftarrow \text{LOCAL SEARCH } (S_{act})$ ;
6:   IF  $S_{act}$  is better than  $S_{best}$  THEN
7:      $S_{best} \leftarrow S_{act}$ ;
8:      $I = 0$ ;
9:   ELSE
10:     $I \leftarrow I+1$ ;
11:  END IF
12: END WHILE
13: RETURN  $S_{best}$ 
```

---

Figura 1. pseudo código.

#### 4.1 Fase de Localización

En esta fase, se encuentra una nueva configuración de los centros de territorio a partir de la solución encontrada en la iteración anterior. Sin embargo, al principio no tenemos una configuración de los centros de territorio. Una forma de obtener esta configuración inicial es usando una heurística GRASP, propuesta por Caballero- Hernández et al. [3] la cual se utilizó para obtener los centros iniciales para nuestra heurística. Ya en iteraciones posteriores, el cálculo de una nueva configuración de centros es relativamente sencillo: se resuelve un problema 1-mediana para cada territorio. Es decir, para cada territorio, se selecciona un nuevo centro minimizando la suma de las distancias a todos los demás nodos en el territorio, esto es, un  $c_k = \operatorname{argmin}_{i \in V_k} \{\sum_{j \in V_k} d_{ij}\}$  es seleccionado, donde  $V_k$  es el conjunto de nodos el territorio  $k$  y  $c_k$  es el centro del territorio.

Esta es la fase más sencilla del algoritmo. La complejidad es  $O(n^2 p)$ . Sin embargo, la elección de los centros de territorios tiene un impacto considerable en los territorios que se obtengan de

tal manera que una "mala" selección de los centros rara vez dará territorios aceptables.

#### 4.2 Fase de Asignación

La configuración de los centros ubicados en la fase anterior se utiliza como entrada a esta fase. La formulación propuesta Segura-Ramiro [1] se propone para aprovechar la estructura del problema y también para obtener un mejor equilibrio en los territorios. La tolerancia se reduce a 0 en una actividad y las restricciones de enteros en las variables pueden tomar cualquier valor entre cero y uno.

Estas áreas se denominan zonas de división o simplemente splits. Note que esto significa que estas áreas básicas se han asignado a dos o más territorios.

Ahora es necesario asignar las variables fraccionarias de cada división a un solo centro. Así que el problema es cómo asignar estas variables a fin de mantener el equilibrio perfecto tanto como sea posible. Este problema se llama el problema de resolución dividida y la metodología diseñada para resolverlo se explica en la siguiente sección.

#### 4.2.1 Resolución de Nodos División

Para cada división, es necesario decidir a qué territorio o centro se asignara el nodo.

En nuestro caso, los criterios siguientes se utilizan para decidir cuál será el territorio de cada nodo de división. Los criterios que se evalúan en el orden en que aparecen:

1. Reparar la conectividad: a fin de reparar un posible territorio inconexo, primero se busca insertar el nodo de división en un territorio inconexo donde la incorporación de este nodo repare la conectividad.
2. Balance: buscar cumplir con el criterio de balanceo determinando que territorio cumpliría dicho criterio.

3. Compacidad: Una medida de impacto se calcula con el fin de elegir el territorio que ofrezca una menor compacidad.

#### 4.3 Búsqueda Local

Después de la fase de localización se procede a aplicarle un procedimiento de mejora que consiste en una búsqueda local propuesto por Ríos-Mercado [4]. Los movimientos en esta búsqueda local consisten en mover una unidad básica de un territorio a otra como se indica en la Figura 2.

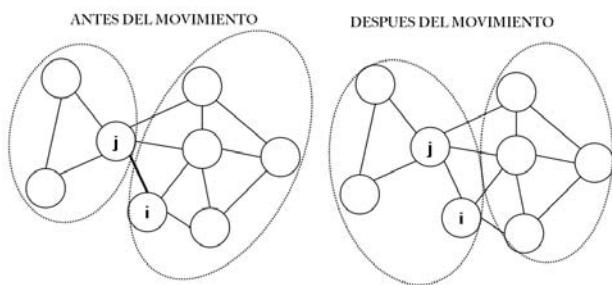


Figura 2. Movimiento de un territorio a otro.

Una vecindad  $N(S)$  se compone de todas las soluciones alcanzables desde  $S$  al mover una unidad básica  $i$  de su territorio actual  $t(i)$  a un vecino territorio  $t(j)$ , donde  $j$  es la unidad básica en el territorio correspondiente  $t(j)$  al lado a  $i$ , sin crear una solución no contiguos. Tal movimiento se denota por  $move(i, j)$  y se ilustra en la figura, donde  $move(i, j)$  está representada por el arco  $(i, j)$ . Tenga en cuenta que  $move(i, j)$  sólo es posible si  $V_{t(j)} \cup \{i\}$  están conectados (que es siempre el caso si el arco  $(i, j)$  existe), y  $V_{t(j)} / \{i\}$  permanece conectado. En la práctica un criterio de parada adicional, como `limit_moves`, se añade para evitar realizar la búsqueda durante una cantidad relativamente grande de tiempo. Por lo que el procedimiento se detiene tan pronto como encuentre óptimos locales o el número de movimientos excede `limit_moves`.

#### **5. Resultados Computacionales**

Para poner a prueba el procedimiento propuesto, las instancias se generaron al azar utilizando varios parámetros que se comentaran posteriormente. Todos los procedimientos fueron construidos en el lenguaje de programación C# y compilado con Visual Studio 2010. Los experimentos se realizaron en un equipo Intel Core i7 con Windows 7 como sistema operativo. El Optimizador Solver Microsoft Foundation versión académica se utilizó cuando fue necesario por el procedimiento.

Con el fin de profundizar en el desempeño del procedimiento, se han considerado varios aspectos: tiempo de ejecución, el número de divisiones (splits), la factibilidad en el balanceo y la conectividad. El procedimiento se llevó a cabo en 2 etapas:

- Ajuste de parámetros para el procedimiento de localización-asignación
- Comportamiento en diferentes combinaciones de factores que influyen en el diseño territorial

#### 5.1 Generador de Instancias

Un generador de instancias aleatorias fue diseñado para generar bloques distribuidos geográficamente. En la vida real, los bloques se agrupan en "áreas fundamentales". La experimentación se llevó a cabo con áreas de 500, 1000 y 2000, que se generaron de acuerdo a su plano cartesiano ( $x, y$ ). Estas coordenadas, fueron generadas al azar y de manera uniforme en el rango [500, 500]. El procedimiento de generación de las instancias está construido de tal manera que el conjunto de nodos generados forman un grafo plano, es decir, ninguna arista puede cruzar a otra arista del grafo. Los nodos están siempre conectados de manera que el gráfico resultante sea un grafo Gabriel. Para la asignación de las actividades (el número de clientes y el volumen de las ventas) se distribuye

uniformemente entre 0 y 3 para el número de clientes por bloque, y entre 1 y 12 para el volumen de ventas. Las instancias fueron probadas con los siguientes parámetros para analizar el comportamiento del algoritmo:

- Número de territorios para construir ( $p$ ): 20, 40 y 60
- La tolerancia sobre el balance ( $\tau$ ): 5%

## 5.2 Parámetros

Los siguientes parámetros fueron evaluados:

- $\lambda$  El parámetro utilizado en la función de mérito para la búsqueda local. Véase Tabla 1.
- $m$  Criterio de parada para el algoritmo. Establecido en 100. Véase la figura 3.

Tamaño	Valores de $\lambda$										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
20	1	1	1	1	0	2	3	1	6	13	0
40	0	0	2	1	1	2	0	6	17	1	0
60	0	1	0	1	2	1	4	13	8	0	0

Tabla 1. La calibración de  $\lambda$

Para la calibración de  $\lambda$  se probaron los valores de 0 a 1 en intervalos de 0.1, estos valores se utilizaron para resolver 10 casos por cada combinación (número de territorios y el número de unidades básicas) para un total de 990 casos. La Tabla 1 muestra el número de veces que se ha alcanzado la mejor solución con la  $\lambda$  y se encontró que el mejor valor  $\lambda$  varía principalmente con el número de territorios. Para las instancias de 20 territorios utilizó  $\lambda = 0.9$ , para las instancias de 40 territorios utilizó  $\lambda = 0.8$  y, finalmente, las instancias de los 60 territorios se utilizó  $\lambda = 0.7$ .

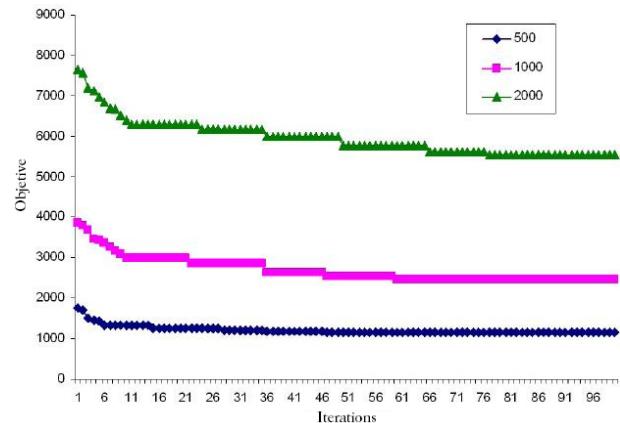


Figura 3. Valor de la función objetivo durante la ejecución

En la Figura 3, se muestra el comportamiento de la función objetivo a lo largo de todo el proceso. El valor de  $m$  (nuestro criterio de parada) se fijó a 100, ya que se puede ver fácilmente que la mejora se consigue poco más allá de este límite.

## 5.3 Evaluación del Procedimiento

Varios aspectos fueron probados con el fin de comprender mejor el comportamiento del procedimiento de búsqueda local: tiempo de ejecución, el número de iteraciones, nodos de división y factibilidad.

### 5.3.1 Tiempo de Ejecución

Dos factores se consideran para medir el tiempo de ejecución: el tamaño de la instancia ( $n$ ) y el número de territorios ( $p$ ) con el fin de comprobar hasta qué punto estos factores influyen en el tiempo de ejecución. La Figura 4 muestra los resultados de este experimento. Como era de esperar, el tiempo de ejecución crece a medida que la complejidad crece (medida por el número de territorios y el número de nodos). La Figura 4 muestra el resumen de los tiempos de ejecución para cada tamaño de la instancia y el número de territorios.

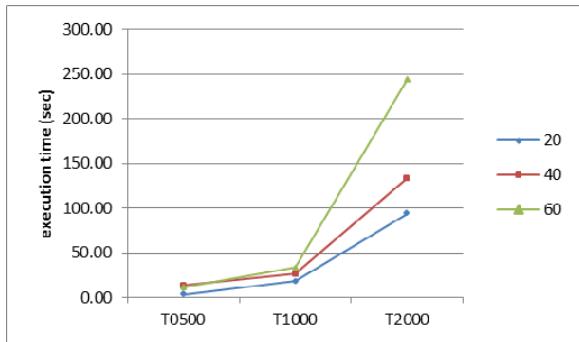


Figura 4. Tiempo promedio de ejecución teniendo en cuenta el tamaño de la instancia (T) y el tamaño del territorio p

### 5.3.2 Número de Iteraciones

Al igual que en tiempo de ejecución, el número de territorios  $p$  y el tamaño de la instancia se consideran para verificar el número de iteraciones necesarias para encontrar la mejor solución. La Figura 5 muestra el número medio de iteraciones, tomando los valores de  $p$  como 20, 40 y 60, cada una de las series mostrada en la Figura 5 representa el tamaño de la instancia. Los resultados que se muestran en esta figura se asemejan a los de la Figura 5 y se llegan a conclusiones similares.

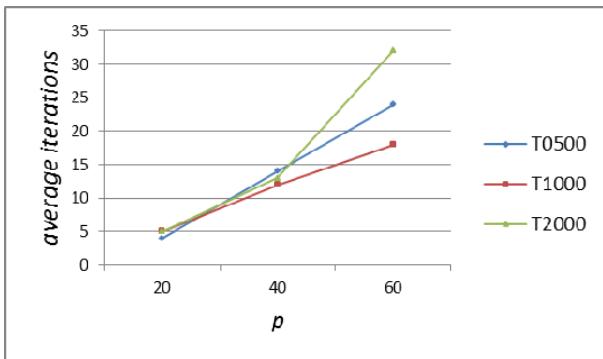


Figura 6. Promedio del número de iteraciones

### 5.3.3 Nodos División

En la Tabla 2 mostramos el mejor caso, el peor caso y un promedio en el número de nodos de división agrupados por el número de nodos de la instancia y número de territorios, como se puede observar en el Tabla 2, el principal factor que afecta

el número de nodos de división es el número de territorios  $p$ .

		T0500	T1000	T2000
$p=20$	Mejor	23.00	28.00	44.00
	Promedio	27.70	37.45	56.10
	Peor	36.00	50.00	85.00
$p=40$	Mejor	45.00	56.00	75.00
	Promedio	49.05	60.90	90.25
	Peor	53.00	68.00	114.00
$p=60$	Mejor	62.00	79.00	102.00
	Promedio	67.35	83.45	116.70
	Peor	73.00	90.00	142.00

Tabla 2. Número de divisiones de las diferentes instancias.

### 5.3.4 Análisis de Factibilidad

El espíritu de la heurística propuesta es tratar de reparar la conexidad siempre que sea posible y buscar la factibilidad con respecto a las restricciones de equilibrio y la conectividad.

En cuanto a la conectividad, el procedimiento de resolución de nodos de división no garantiza una solución factible con respecto a el equilibrio y la conectividad. La Tabla 3 muestra el porcentaje de veces que el procedimiento pudo encontrar una solución factible en la experimentación.

<b>P</b>	<b>T500</b>	<b>T1000</b>	<b>T2000</b>
<b>20</b>	100%	100%	100%
<b>40</b>	100%	100%	100%
<b>60</b>	100%	100%	100%

Tabla 3. Porcentaje de soluciones factibles.

(Conectividad)

Ahora bien, dado que se encontró que el algoritmo en general toma en el orden de 50 a 60 iteraciones para finalizar, la probabilidad de éxito de la ejecución general del algoritmo es prácticamente 1. Esto se puede ver en la Tabla 4.

<b>p</b>	<b>T500</b>	<b>T1000</b>	<b>T2000</b>
20	100%	100%	100%
40	85%	100%	100%
60	0%	85%	100%

Tabla 4. Porcentaje de soluciones factibles.

(Balance)

## Referencias

- [1] Segura Ramiro, J. A. (2008) *Un Algoritmo de Localización-Asignación para el Diseño Eficiente de Planes Territoriales de Uso Comercial*. San Nicolás de los Garza, N. L., México: Tesis de Maestría, Facultad de Ingeniería Mecánica y Eléctrica, U.A.N.L.
- [2] Kalcsics J., Nickel S. y Schröder. M. (2005) *Toward a unified territorial design approach: Applications, algorithms, and GIS integration*. *TOP*, 13(1):1-74,
- [3] Caballero-Hernández S. I., Ríos Mercado R. Z. y F. López. (2005) *Solución heurística a un problema de diseño de territorios comerciales con restricciones de asignación conjunta mediante GRASP*. Zaragoza España: En Memorias del II Congreso Español de Informática.
- [4] Ríos-Mercado R. Z. (2007) *Computational experience with a reactive GRASP for a large scale commercial territory design problem*. Stuttgart, Alemania: En M. J. Geiger y W. Habenicht, editores, *Proceedings of EU/ME 2007 Metaheuristics in the Service Industry*, pp. 72–79, ISBN: 978-3-00-022976-3,