

# Proyecto Metaheurísticas: Fase 1

Andrés Felipe Romero Silva<sup>1</sup>, Sebastián Alfonso Bueno Patiño<sup>2</sup>

<sup>1</sup> Universidad de los Andes  
Cra. 1 #18a-12, Bogotá, Bogotá D.C.  
af.romero11@uniandes.edu.co

<sup>2</sup> Universidad de los Andes  
Cra. 1 #18a-12, Bogotá, Bogotá D.C.  
sa.bueno@uniandes.edu.co

## Resumen

El presente documento aborda el desarrollo de la primera fase del proyecto del curso de Metaheurísticas ofrecido por el Departamento de Ingeniería Industrial de la Universidad de los Andes, en la cual se modela el problema en cuestión y se diseña, implementa y valida una heurística desarrollada por la pareja de estudiantes.

## 1 Introducción

El proyecto del curso Metaheurísticas consiste en la implementación, experimentación, análisis de resultados obtenidos y elaboración de informe de una metodología de optimización para el problema del comprador viajero (TPP, de sus siglas en inglés, *Traveling Purchaser Problem*).

Se espera que el proyecto conserve un diseño metodológico donde se especifique claramente la descripción del problema, el esquema de optimización, el cual comprende el modelo matemático, pseudocódigo, etc., junto a las ventajas y desventajas de este. Además de esto, se debe implementar la metodología y se debe realizar la experimentación y validación del desarrollo. Sumado a la experimentación se debe adicionar un análisis de los resultados obtenidos con relación a las soluciones alcanzadas.

En específico, la Fase 1 del proyecto comprende el modelado del problema del comprador viajero, además del diseño, la implementación y la validación de una heurística propuesta por los autores del presente trabajo.

## 2 Descripción del problema

### 2.1 Problema del comprador viajero (TPP)

Este problema consiste en establecer un plan de compra de productos en un conjunto de proveedores disponibles, entendidos como los puntos de venta, minimizando los costos de transporte asociados a la ruta que sigue el camión, en el cual se almacenan los productos comprados, y los costos de adquisición de los productos, entendido este costo como el precio de venta por producto adquirido.

### 2.2 Modelo matemático del problema

El problema TPP que vamos a solucionar es una versión simplificada conocida como UTPP, el cuál consiste en

que todos los mercados tienen suficiente producto para suplir la demanda. Lo anterior simplifica el número de restricciones y los parámetros de entrada del problema dando como resultado el siguiente modelo:

#### I. Conjuntos

Mercados:  $M = \{ 0, 1, 2, 3, 4, \dots, n \}$

Productos:  $P = \{ 0, 1, 2, 3, 4, \dots, n \}$

#### II. Parámetros

$c_{ij}$  = costo de transporte entre el market  $i, j \in E^2$

$b_{ij}$  = costo de comprar el producto  $i \in P$  en el market  $j \in M$

$o_{ij}$  = binaria, disponibilidad del producto  $i \in P$  en el market  $j \in M$

#### III. Variables

$x_{ij}$  = binaria, 1 si el arco entre  $i, j \in M^2$  es elegido, 0 d.l.c.

$y_{ij}$  = binaria, 1 si el producto  $i \in P$  se compra en  $j \in M$ , 0 d.l.c.

$z_{ijp}$  = binaria, multiplicación entre  $y_{ij}x_{pj}$   $i, j \in M^2, p \in P$ .

#### IV. Restricciones

a) Todos los productos se tienen que comprar

$$\sum_{j \in M} y_{ij} o_{ij} = 1 \quad \forall i \in P$$

b) Balance de masa en los nodos de las rutas

$$x_{ij} - x_{ji} = 0 \quad \forall i, j \in M^2$$

c) Las rutas deben pasar y volver al deposito

$$\sum_{j \in M} x_{0j} = 1$$

$$\sum_{j \in M} x_{j0} = 1$$

d) Para poder comprar en un mercado, hay que visitarlo en una ruta

$$z_{ijp} = y_{ip} \quad \forall p \in P; (i, j) \in M^2$$

e) Linealización de la multiplicación de variables binarias

$$z_{ijp} \geq x_{ij} + y_{pj} - 1 \quad \forall p \in P; (i, j) \in M^2$$

$$z_{ijp} \leq x_{ij} \quad \forall p \in P; (i, j) \in M^2$$

$$z_{ijp} \leq y_{pj} \quad \forall p \in P; (i, j) \in M^2$$

f) Restricciones de eliminación de subtours

Hay muchas maneras de hacer este tipo de restricciones, o se podría manejar una optimización dinámica, por ejemplo, cortes. Que elimine los subtours a medida que se generan. Estas restricciones aumentan exponencialmente y hacen que el problema sea NP-Hard

#### V. Función Objetivo

$$\min \sum_{i \in M} \sum_{j \in M} x_{ij} c_{ij} + \sum_{i \in P} \sum_{j \in M} y_{ij} b_{ij}$$

## 2.3 Estado del arte

El problema UTPP fué introducido por primera vez en 1966 (Burstall, 1966) como un problema de *shceduling* y posteriormente se redefinió como un problema de transporte. Desde entonces se han explorado numerosas metodologías de solución.

### 2.3.1 Metodologías exactas

Autor	Metodología	Aporte
Buzacott y Dutta (1971)	Optimización exacta	Programación de la máquina para la fabricación de tubos de acero
Padberg y Rinaldi (1991)	Optimización exacta	Separación dinámica de restricciones de subtours
Singh y Van Oudheusden (1997)	Branch-and-bound	División del conjunto de todos los recorridos posibles en subconjuntos más pequeños
Laporte et al. (2003)	Branch-and-bound	Estudio poliédrico del problema que combina los politopos de ciclo y set-covering

### 2.3.2. Heurísticas

Las heurísticas desarrolladas para este problema se basan, en su mayoría, en el concepto de “ahorros” el que consiste en ahorrar costos netos mediante la disminución en el costo de comprar y aumentar el costo de transporte. Algunas heurísticas constructivas son:

1. **Generalized Savings Heuristic (GSH) Golden et al. (1981):** Comienza desde un recorrido inicial que contiene solo el depósito, y que agrega un proveedor en cada iteración.
2. **Tour Reduction Heuristic (TRH) Ong (1982):** comienza con un recorrido que involucra a un subconjunto de proveedores que satisfacen los productos demanda, e iterativamente elimina un proveedor produciendo la máxima reducción de los costos totales.
3. **Commodity Adding Heuristic (CAH) Pearn (1991):** comienza con una solución inicial que contiene solo el depósito y el proveedor que minimiza el costo total de compra del primer producto. En cada iteración se evalúa la conveniencia de comprar el siguiente producto en un proveedor del tour o agregar un nuevo proveedor al tour.

### 2.3.3. Metaheurísticas

En este campo se han probado distintas metodologías como *Tabu Search* por Voß (1996) , El-Dean (2008) y Mansini et al. (2005). También metodologías tipo GRASP por Ochi et al. (2001), adicionalmente Ochi et al. (2001) también exploró metodologías de VNS y GRASP+VNS. Bontoux and Feillet

(2008) exploraron metodologías basadas en colonia de hormigas y Ochi et al. (1997) también evaluó algoritmos genéticos (GA)

### 3 Metodología

#### 3.1 Presentación de la heurística propuesta como solución del problema TPP

La heurística considera un número determinado de *markets*, supermercados o puntos de venta, a los cuales el comprador viajante se puede dirigir para comprar los productos. En primer lugar, la heurística asume el supuesto de que no se debe comprar ninguno de los productos. Es decir, abordar el problema inicialmente como un TSP (de sus siglas en inglés, *Traveling Salesman Problem*). De tal forma que se recorran los puntos de venta únicamente considerando la búsqueda del camino más corto, entendido como la ruta menos costosa en términos de costos de desplazamiento. Se pretende encontrar esta ruta de tal forma que el camión visite todos los puntos de venta. Ahora bien, la regla de decisión de comprar un producto determinado reside en comprarlo en aquel punto de venta por el cual el camión pasa en el que el producto tiene un menor precio.

Posteriormente, y después de la selección inicial de la ruta y los puntos de venta en los cuales se compra preliminarmente cada producto bajo la regla de decisión descrita, se eliminan iterativamente ciertos puntos de venta, es decir, uno a la vez. Se evalúan todos los puntos de venta que se visitan hasta el momento (téngase en cuenta que inicialmente se visitan todos los puntos de venta) y hallar el valor de la función objetivo tras la eliminación de cada uno de los puntos de venta (escenarios individuales). La regla de decisión para eliminar un punto de venta reside en eliminar aquel punto de venta cuya eliminación se traduce en una menor función objetivo (menores costos totales). Nuevamente, con los puntos de venta restantes tras la eliminación de uno de ellos, se vuelve a aplicar la regla de decisión para los puntos de venta que aún se visitan. Así sucesivamente hasta obtener un número de puntos de venta que se visitan igual al número de productos que se deben comprar.

En cada iteración, y dada la eliminación de un punto de venta determinado, puede ser necesario actualizar el punto de venta en el cual se compra el producto, atendiendo a la regla de decisión de compra de los productos, es decir, buscando nuevamente cuál es el punto de venta entre los visitados en el cual el producto se adquiere a un menor precio.

Es importante anotar que, una vez se obtiene un número de puntos de venta que se visitan igual al número de productos que se deben comprar, pero se encuentra que no en todos los puntos de venta se compran productos, se eliminan precisamente aquellos en los cuales no hay compra de ningún producto. Esto puede presentarse dado el caso en el que dos productos son obtenidos a un menor precio en un mismo punto de venta.

### 3.2 Pseudocódigo

La heurística constructiva que se desarrolló en el proyecto está basada en el método **Tour Reduction Heuristic (TRH) propuesto por Ong (1982)**. Adicionalmente, se separa el problema en 2 subproblemas: el problema de ruteo (TSP) y el problema de decisión de en dónde comprar cada uno de los productos. De esta manera si se tiene el TSP, la decisión de comprar es trivial. Pues siempre se va a preferir comprar en el market, que se encuentre dentro del tour, que ofrezca este producto al menor precio.

```

function constructive(markets, products, cost_matrix, offer_section):
1.
2.  route ← nearestNeighbour(markets, cost_matrix) Initialization
3.  route ← 2opt(route, cost_matrix) Initial route improvement
4.  buys ← whereToBuy(route, products, offer_section) Trivial decision
5.  objective_function ← totalCosts(route, buys)
6.  improve ← True
7.    while Improve:
8.      improve ← False Out criterion
9.      newRoute ← bestElimination(route)
10.     newBuys ← whereToBuy(route, products, offer_section)
11.     newFO ← totalCosts(newRoute, newBuys)
12.
13.     if newFO < objective_function:
14.       improve ← True
15.       route ← upgradeSolutionRoute(route, newRoute)
16.       buys ← upgradeSolutionBuys(buys, newBuys)
17.
18.  return route, buys

```

## 4 Experimentos y análisis de resultados

### 4.1 Implementación de la heurística en lenguaje Python

La heurística completa está implementada y debidamente documentada en el archivo anexo *constructive.py*

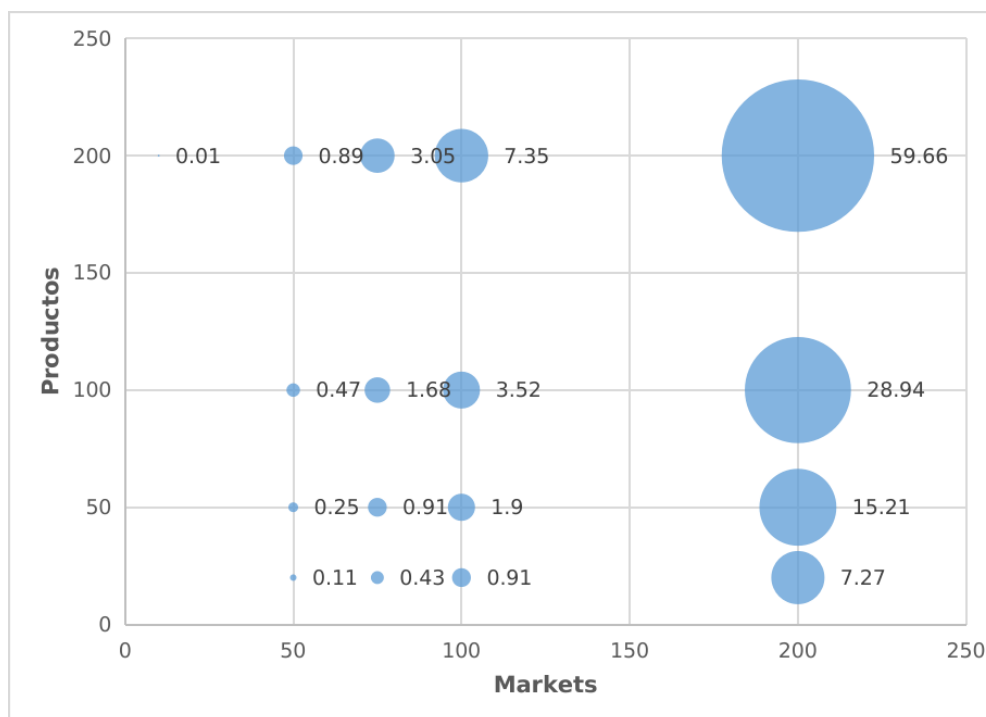
### 4.2 Validación de la heurística propuesta

La heurística se validó en un total de 42 instancias. Esta logró generar soluciones factibles para todas las instancias evaluadas. Los resultados obtenidos se muestran en la siguiente tabla.

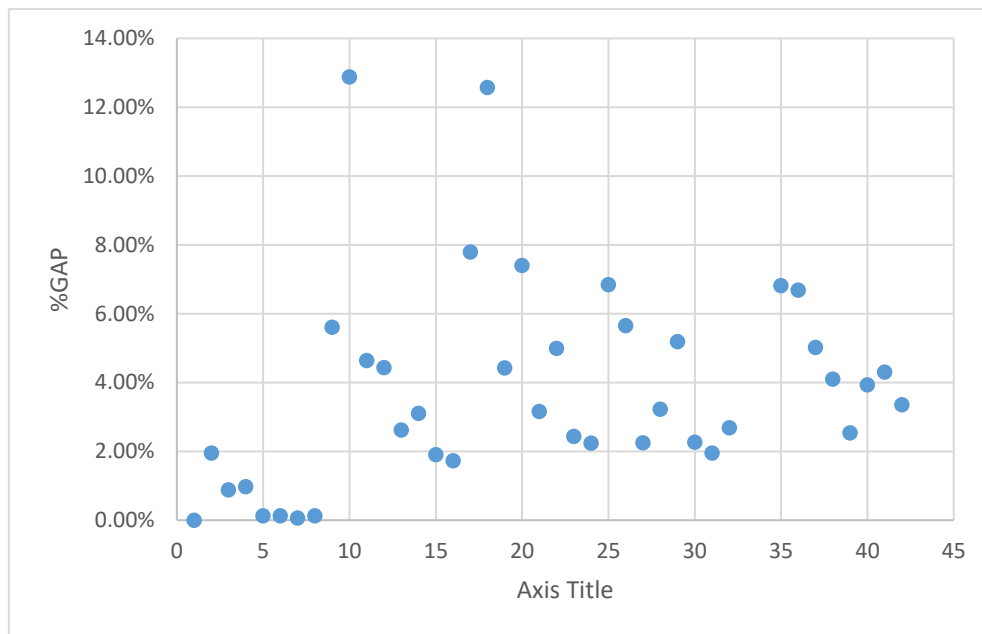
**Tabla 1: Resultados obtenidos mediante la heurística constructiva**

Markets	Productos	Versión	Optimo	Solución	Tiempo[s]	GAP
10	20	2	546	546	0	0.00%
10	20	3	563	574	0	1.95%
10	50	2	1244	1255	0	0.88%
10	50	3	1229	1241	0	0.98%
10	100	2	2365	2368	0	0.13%
10	100	3	2368	2371	0	0.13%

10	200	2	4555	4558	0.01	0.07%
10	200	3	4583	4589	0	0.13%
50	20	2	517	546	0.11	5.61%
50	20	3	520	587	0.16	12.88%
50	50	2	1229	1286	0.25	4.64%
50	50	3	1239	1294	0.3	4.44%
50	100	2	2325	2386	0.47	2.62%
50	100	3	2319	2391	0.52	3.10%
50	200	2	4454	4539	0.89	1.91%
50	200	3	4462	4539	0.93	1.73%
75	20	2	526	567	0.43	7.79%
75	20	3	525	591	0.4	12.57%
75	50	2	1220	1274	0.91	4.43%
75	50	3	1243	1335	0.9	7.40%
75	100	2	2309	2382	1.68	3.16%
75	100	3	2304	2419	1.62	4.99%
75	200	2	4468	4577	3.05	2.44%
75	200	3	4462	4562	2.95	2.24%
100	20	2	511	546	0.91	6.85%
100	20	3	495	523	0.91	5.66%
100	50	2	1155	1181	1.9	2.25%
100	50	3	1146	1183	1.93	3.23%
100	100	2	2292	2411	3.52	5.19%
100	100	3	2296	2348	3.86	2.26%
100	200	2	4457	4544	7.35	1.95%
100	200	3	4430	4549	7.55	2.69%



**Ilustración 1: Tiempo de ejecución para cada una de las instancias**



**Ilustración 2: Distribución del % de GAP obtenido para cada las instancias evaluadas**

### 4.3 Análisis de resultados

Se evidencia que el tiempo de ejecución del constructivo aumenta al aumentar tanto el número de markets como el número de productos demandados. Por otra parte, se obtuvo en promedio un GAP de 3.83% para las instancias evaluadas y que tienen solución óptima conocida, también se obtuvo para las instancias AsimSingh.300.150.5 y AsimSingh.300.200.1 un GAP de 4.3% y 3.4% con respecto a los BKS actuales. Lo cual indica que el constructivo es bastante bueno y alcanza muy buenas soluciones iniciales y podría indicar que, en un futuro, al adicionar una metaheurística que gobierne dicho constructor no se necesiten muchas iteraciones para obtener mejores soluciones.

## 5 Conclusiones

Pendiente para Fase 2.

## Referencias

Pendiente para Fase 2.