# Un algoritmo genético con llaves aleatorias con sesgo para el problema del viajero familiar multi-agente capacitado.

Saúl Domínguez Casasola[1*], José Luis González Velarde[1], y Yasmín Á. Ríos Solís[1].
[1]Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Monterrey, Nuevo León.
E. Garza Sada 2501 Sur, C.P. 64849, Monterrey, Nuevo León, México.
*Correo electrónico: A00988197@itesm.mx

## RESUMEN

La motivación del problema del viajero familiar multi-agente capacitado (CFTSP-*Capacitated Family Travel Salespersons Problem*), son los almacenes con ubicaciones multi-SKU y productos iguales en ubicaciones distintas. Esto lleva a decidir: las ubicaciones a visitar, el orden de las visitas y su asignación a los agentes. Matemáticamente, se considera un grafo completo, cuyos nodos están divididos en familias disjuntas, el CFTSP consiste en encontrar un subconjunto de nodos a visitar, minimizando la distancia total recorrida. Se debe satisfacer la demanda para cada familia sin exceder la capacidad de los agentes. Para este problema, se propone un modelo de programación entera; sin embargo, para algunas instancias no se consiguen soluciones factibles en un tiempo de cómputo razonable. Por ello, se utiliza un algoritmo genético con llaves aleatorias con sesgo (BRKGA – *Biased Random Key Genetic Algorithm*), que consigue obtener buenos resultados. Finalmente, se comparan ambas estrategias, y se hacen conclusiones.

**Palabras clave**: Optimización Combinatoria, BRKGA, CFTSP, Logística.

## INTRODUCCIÓN

Tecnologías como la radiofrecuencia permiten a los almacenes tener un mismo tipo de producto en diferentes ubicaciones, así como ubicaciones con distintos tipos de productos. La intención es tener rutas para los agentes en el surtido, que minimicen la distancia total que se recorre, satisfaciendo los pedidos, y respetando las capacidades de carga. Para ello, se formuló un modelo matemático; sin embargo, la naturaleza combinatoria del problema en algunas instancias no permite obtener soluciones óptimas en tiempos razonables. Como alternativa se propone un BRKGA que, consigue buenos resultados, en tiempos de cómputo menores.

## CFTSP

Sea un grafo completo $G = (\mathbf{N} \cup \{0\}, E)$ donde $0$ es el depósito y $\mathbf{N}$ es el conjunto de nodos particionado en $L$ familias disjuntas. Cada familia $F_l$ contiene $f_l$ nodos con un peso asociado $w_l$, de los cuales deben visitarse $v_l \le f_l$. En el conjunto de aristas $E$ están indexadas las distancias $d_{ij} > 0$ entre los nodos $i, j$, $\forall i \neq j \in \{0, \cup\, \mathbf{N}\}$. Además, se tiene un conjunto $\mathbf{P} = \{1, \dots, P\}$ de agentes con capacidad $c$. El problema es encontrar las rutas $(V_1, V_2, \dots, V_P)$ en $G$, con $\sum_{i \in V_p} w_i \le c$ para $p = 1, \dots, m$ y $|(V_1 \cup V_2 \cup V_3 \dots V_{m-1} \cup V_m) \cap F_l| = v_l$ al tiempo que se minimiza:

$$\sum_{d_{ij} \in \delta(V_1, V_2, \dots, V_m)} d_{ij}$$

Donde $\delta(V_1, V_2, \dots, V_m)$ denota el conjunto de aristas que se recorren al elegir las rutas $(V_1, V_2, \dots, V_m)$. Cuando $\mathbf{P} = \{1\}$, $\sum_{i \in \mathbf{N}} w_i \le c$ y $\sum_{l=1}^{L} v_l = |\mathbf{N}|$ el CFTSP se reduce al Problema del Agente Viajero (TSP por sus siglas en inglés *Traveling Salesman Problem*).

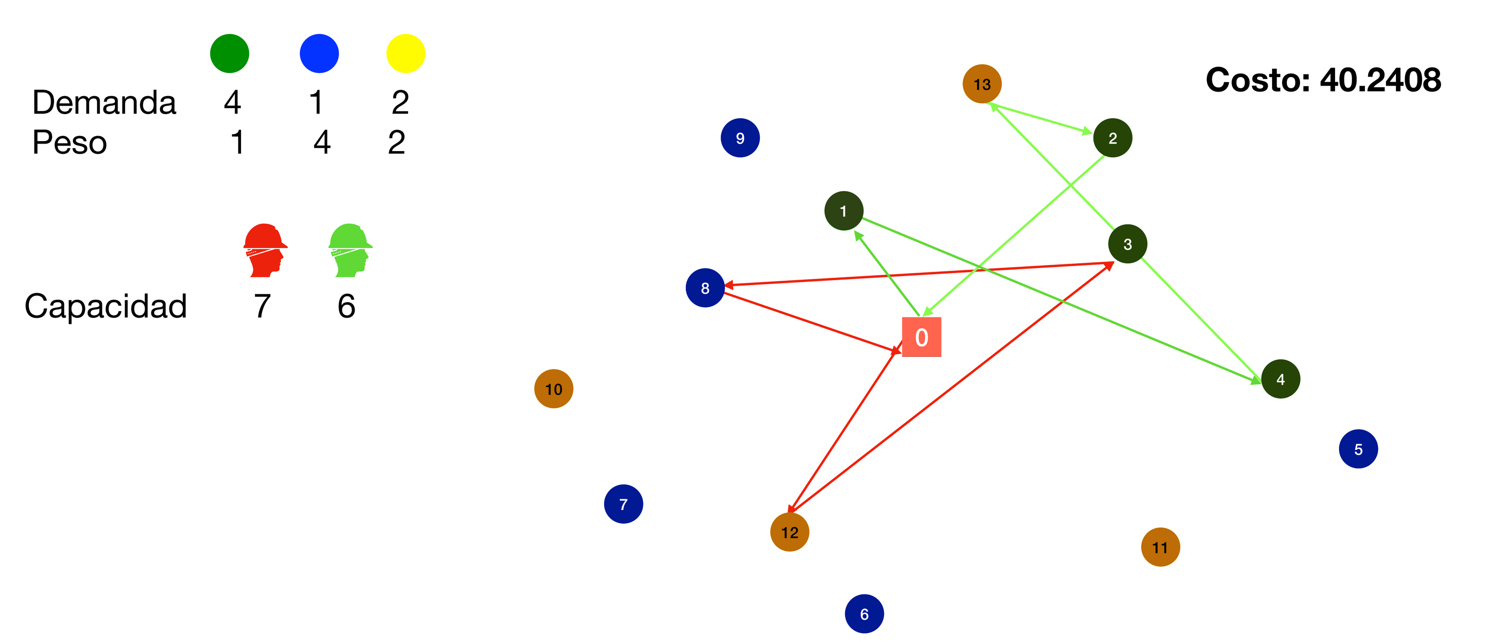


**Figura 01**. Ejemplo de solución para el CFTSP.

## RESULTADOS

| | | | | | Modelo | | | BRKGA | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Instancia** | **N** | **KN** | **K** | **c** | **Cost** | **Time** | **GAP** | **Promedio** | **Mejor Solución** | **D.E.** | **C.V.** | **Tiempo** | **Iteraciones** |
| SYBurma1001A | 13 | 6 | 2 | 3 | **15.25** | 0 | 0% | **15.25** | **15.25** | 0.000 | 0.000 | 2 | 105 |
| SYBurma1001B | 13 | 6 | 3 | 2 | **18.23** | 0 | 0% | **18.23** | **18.23** | 0.000 | 0.000 | 2 | 108 |
| SYBurma1002A | 13 | 10 | 2 | 5 | **32.51** | 2 | 0% | **32.51** | **32.51** | 0.000 | 0.000 | 3 | 112 |
| SYBurma1002B | 13 | 10 | 5 | 2 | **48.94** | 0 | 0% | **48.94** | **48.94** | 0.000 | 0.000 | 3 | 126 |
| SYBurma1003AB | 13 | 4 | 2 | 2 | **13.63** | 0 | 0% | **13.63** | **13.63** | 0.000 | 0.000 | 2 | 102 |
| SYBayg1001AB | 28 | 16 | 4 | 4 | **8,304.87** | 435 | 0% | 8,481.87 | **8,304.87** | 140.149 | 0.017 | 7 | 179 |
| SYBayg1002A | 28 | 17 | 4 | 5 | **8,311.32** | 2458 | 0.01% | 8,517.99 | 8,407.47 | 108.070 | 0.013 | 7 | 156 |
| SYBayg1002B | 28 | 17 | 5 | 4 | **9,131.59** | 3600 | 3% | 9,423.32 | 9,363.80 | 133.078 | 0.014 | 7 | 161 |
| SYBayg1003A | 28 | 18 | 4 | 5 | **7,687.73** | 3600 | 1% | 7,836.93 | 7,721.19 | 133.476 | 0.017 | 8 | 188 |
| SYBayg1003B | 28 | 18 | 5 | 4 | **8,457.12** | 3600 | 5% | 8,692.06 | 8,477.12 | 196.342 | 0.023 | 6 | 144 |
| SYAtt1001A | 47 | 34 | 5 | 7 | 45,317.86 | 3600 | 41.81% | 43,719.40 | **42,333.78** | 852.22 | 0.019 | 21 | 327 |
| SYAtt1001B | 47 | 34 | 7 | 5 | 60,246.63 | 3600 | 49.83% | 53,137.82 | **52,529.92** | 552.61 | 0.010 | 22 | 310 |
| SYAtt1003A | 47 | 15 | 5 | 3 | 20,577.77 | 3600 | 26.67% | **20,097.25** | **20,097.25** | 0.00 | 0.000 | 9 | 181 |
| SYAtt1003B | 47 | 15 | 3 | 5 | 15,467.94 | 3600 | 36.10% | 15,162.37 | **14,859.35** | 399.93 | 0.026 | 11 | 226 |
| SYBier1001AB | 126 | 62 | 8 | 8 | 90,946.26 | 3600 | 64.50% | 61,284.40 | **59,372.46** | 1517.09 | 0.025 | 86 | 670 |

**Tabla 01**. Comparación Modelo vs. BRKGA.

## BRKGA

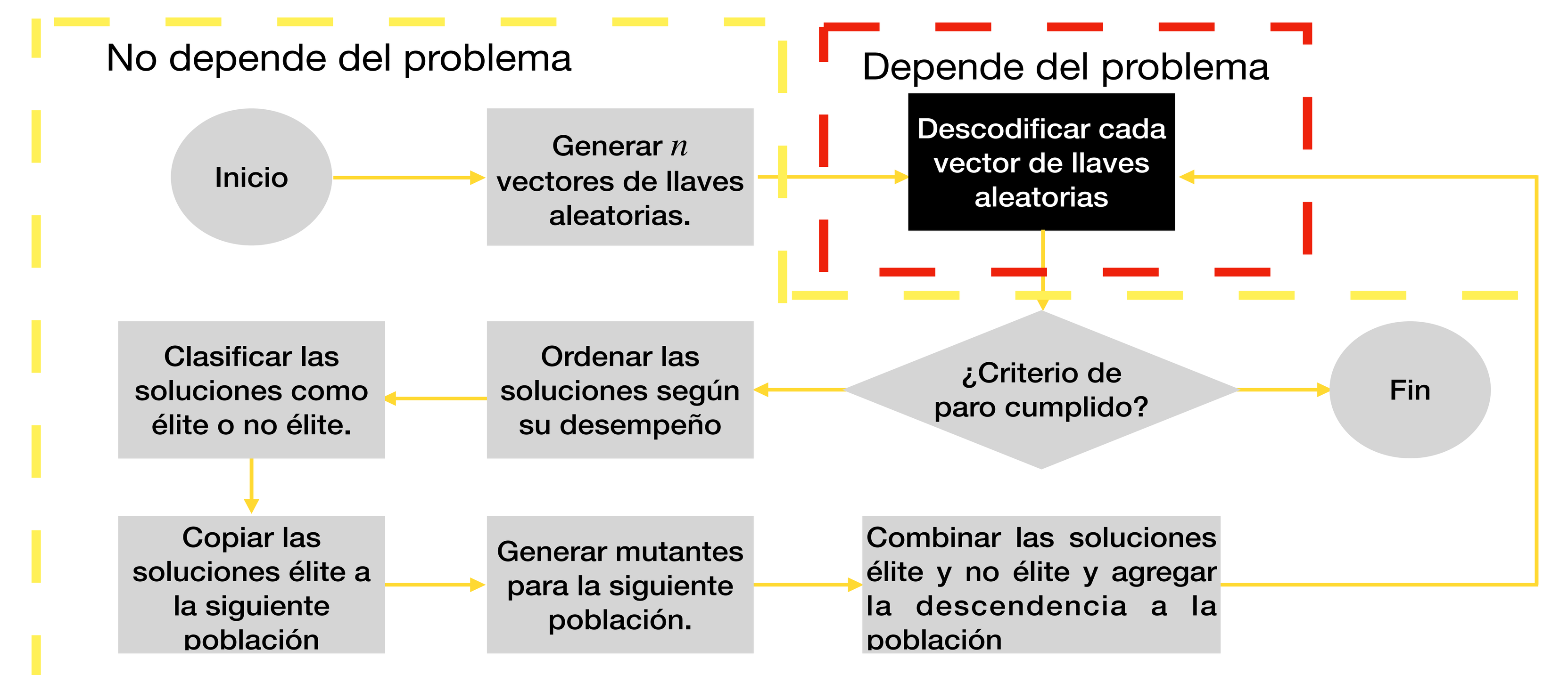


**Figura 02**. Estructura de un BRKGA.

## DESCODIFICADOR

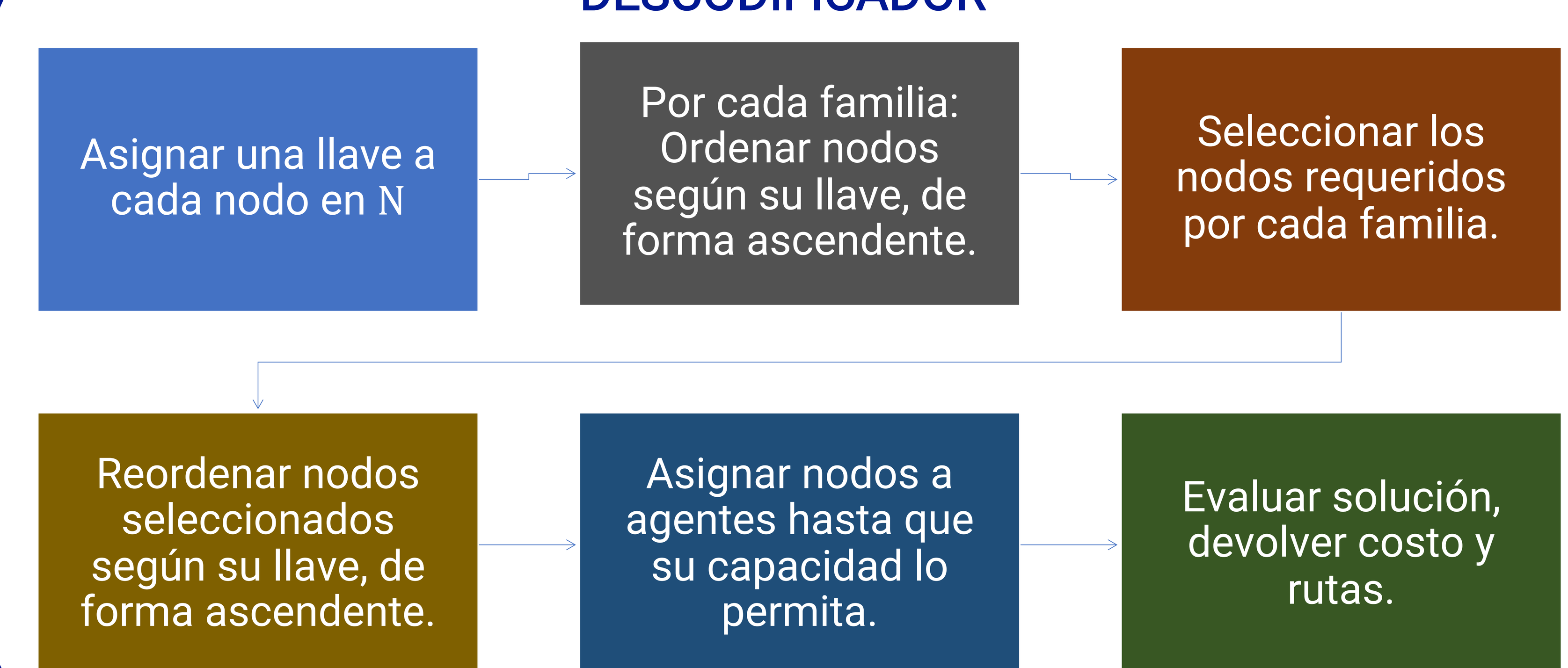


**Figura 03**. Estructura de decodificación.

## EJEMPLO

**Instancia**
$N = 13$ Nodos.
$L = 3$ Familias.
$f_l = [4, 5, 4]$ Nodos por familia.
$v_l = [4, 1, 2]$ Visitas por familia.
$w_l = [1, 1, 1]$ Peso por familia.
$K = 2$ Agentes.
$c = 4$ Capacidad.

**Asignación de llaves**

| Gen | 0.32 | 0.01 | 0.27 | 0.15 | 0.34 | 0.12 | 0.63 | 0.11 | 0.65 | 0.97 | 0.78 | 0.69 | 0.47 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nodo | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Familia | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |

**Selección de nodos a visitar**

| Gen | 0.01 | 0.15 | 0.27 | 0.32 | 0.11 | 0.12 | 0.34 | 0.63 | 0.65 | 0.47 | 0.69 | 0.78 | 0.97 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nodo | 2 | 4 | 3 | 1 | 8 | 6 | 5 | 7 | 9 | 13 | 12 | 11 | 10 |
| Familia | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |

**Selección de rutas**

| Gen | 0.01 | 0.11 | 0.15 | 0.27 | 0.32 | 0.47 | 0.69 |
|---|---|---|---|---|---|---|---|
| Nodo | 2 | 8 | 4 | 3 | 1 | 13 | 12 |

**Asignación a agentes**

[0 -> 2 -> 8 -> 4 -> 3 -> 0] [0 -> 1 -> 13 -> 12 -> 0]

## CONCLUSIONES

Se propone un BRKGA para el CFTSP que se aproxima a los valores óptimos (en los casos conocidos) o incluso los consigue. En instancias de mayor dimensión, mejora las soluciones factibles obtenidas por el método exacto. A menor capacidad de los agentes, se asigna una cantidad mayor de agentes al surtido, y la distancia total recorrida incrementa. Se pueden explorar distintas alternativas para descodificar cromosomas para el BRKGA, así como mecanismos de reparación, en el caso de tener soluciones infactibles.

## REFERENCIAS

1. L. Morán-Mirabal, J. González-Velarde y M. G. Resende, «Randomized heuristics for the family traveling salesperson problem,» International Transactions in Operational Research, vol. 21, nº 1, pp. 41-57, 2014.
2. Ferreira, C. E., Martin, A., de Souza, C. C., Weismantel, R., & Wolsey, L. A. (1998). The node capacitated graph partitioning problem: a computational study. *Mathematical programming*, *81*(2), 229-256.
3. Bernardino, R., & Paias, A. (2018). Solving the family traveling salesman problem. *European Journal of Operational Research*, *267*(2), 453-466.
4. Silva, R. M., Resende, M. G., & Pardalos, P. M. (2015). A Python/C++ library for bound-constrained global optimization using a biased random-key genetic algorithm. *Journal of Combinatorial Optimization*, *30*(3), 710-728.