

Explorando la variabilidad en el tamaño de la población de un algoritmo genético para resolver el problema del agente viajero

Laura Michelle Báez Villegas, Santiago Omar Caballero Morales

Universidad Popular Autónoma del Estado de Puebla,
México

`lauramichele.baez@upaep.edu.mx,`
`santiagoomar.caballero@upaep.mx`

Resumen. El Problema del Agente Viajero (o Travelling Salesman Problem, TSP) es uno de los principales modelos de ruteo dentro del campo logístico. Su complejidad del tipo NP-duro hace que problemas de gran escala sean difíciles de resolver en tiempo razonable. Para ello resulta de mucha utilidad el diseño y la aplicación de algoritmos de búsqueda local o metaheurísticas. En este trabajo, el TSP se aborda mediante el diseño de un Algoritmo Genético que considera una población fija y una población variable como mecanismo para mejorar la convergencia hacia un óptimo local. Los resultados indican que, si bien una población variable puede hacer más rápida la ejecución del Algoritmo Genético y ayuda a evitar que el algoritmo permanezca en un óptimo local de manera prematura, su velocidad de convergencia disminuye con respecto al caso de una población fija (aunque esto involucre un mayor tiempo de ejecución). Esta observación puede ser de utilidad para el ajuste de los parámetros de estos algoritmos para el TSP.

Palabras clave: Problema del agente viajero, algoritmos genéticos, población variable.

Exploring Variability in Population Size of a Genetic Algorithm for Solving the Travelling Salesman Problem

Abstract. The Travelling Salesman Problem (TSP) is one of the main routing models within the logistics field. Its NP-hard complexity makes large-scale problems difficult to solve in a reasonable time. For this, the design and application of local search algorithms or metaheuristics is very useful. In this work, the TSP is approached by designing a Genetic Algorithm that considers a fixed population and a variable population as a mechanism to improve convergence towards a local optimum. The results indicate that, although a variable population can speed up the execution of the Genetic Algorithm and help prevent the algorithm from remaining in a local optimum prematurely, its speed of convergence decreases with respect to the case of a fixed population (although this involves a longer execution time). This observation may be useful for adjusting the parameters of these algorithms for the TSP.

Keywords: Travelling salesman problem, genetic algorithms, variable population.

1. Introducción

El Problema del Agente Viajero, o Travelling Salesman Problem (TSP) es uno de los problemas más importantes dentro de la logística y de la optimización. Este problema modela el escenario de un agente viajero que parte de su ciudad de origen, visita exactamente una vez cada ciudad de un conjunto asignado, y finalmente regresa a su ciudad de origen (punto de partida) [1].

Un recorrido con estas características se conoce como Circuito Hamiltoniano, Ruta o Tour. Por lo tanto, el objetivo del TSP consiste en encontrar el Circuito Hamiltoniano de menor costo o distancia total. La Figura 1 presenta un ejemplo de solución para el TSP con ocho ciudades [1]. Debido a lo anterior, la relevancia del TSP no solo se presenta en los aspectos de distribución, sino también en los traslados dentro de los almacenes y plantas de producción [1].

De igual manera, el TSP es la base de problemas más extensos como el Problema de Ruteo Vehicular o Vehicle Routing Problem (VRP) que puede entenderse como un TSP múltiple, en el cual se tienen restricciones de demanda y capacidad (véase Figura 2) [2, 7]. Si bien el TSP/VRP tienen amplia aplicación en la distribución, la solución de estos es difícil cuando se trata de redes de gran tamaño.

Esto es debido a su complejidad computacional que es del tipo NP-duro, lo cual hace que el tiempo de ejecución de un algoritmo de solución crezca exponencialmente conforme el tamaño del problema crezca linealmente [1, 5, 8]. ¡Por ejemplo, un TSP con 8 ciudades tiene un total de $8! = 40320$ soluciones posibles. ¡En cambio, un problema con sólo una ciudad adicional tiene un total de $9! = 362880$ soluciones posibles.

Por ello, las formulaciones matemáticas mediante programación lineal entera y algoritmos exactos como Branch & Bound pueden generar soluciones en tiempo razonable en redes con menos de 200 nodos, ciudades o ubicaciones [3]. Para obtener soluciones cercanas al óptimo dentro de un tiempo razonable se ha realizado investigación en el campo de los algoritmos de búsqueda local o metaheurísticas.

Para los problemas de ruteo, algoritmos como Búsqueda Tabú (BT) [4], Algoritmos Genéticos (AG) [5], Clarke & Wright (Ahorros) [6], entre otros. En el presente trabajo se considera a los AG para resolver redes del tipo TSP con más de 200 nodos. Si bien esta consideración ya se ha abordado en la literatura especializada, en estos se establecen estructuras fijas para el tamaño de la población, la cual es el medio principal de memoria de soluciones y que, dependiendo de su tamaño, puede hacer más lenta la operación del AG.

Es por esto que el AG propuesto considera una población de tamaño variable con el fin de hacer más rápido el AG, y depurar la población para evitar la convergencia hacia un óptimo local de manera prematura.

Los avances del presente trabajo se describen en las siguientes secciones: en la Sección 2 se presentan las bases de los AG mientras que en la Sección 3 se presentan los detalles del AG propuesto con población variable.

Los resultados se presentan en la Sección 4 y la discusión de estos, incluyendo el trabajo futuro, se presentan en la Sección 5.

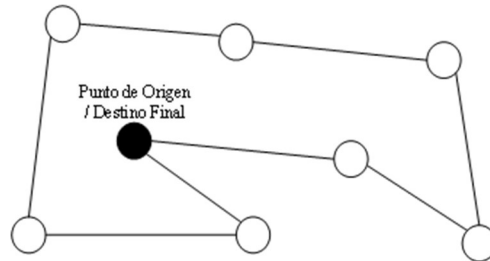


Fig. 1. Ejemplo de Solución para el TSP (basado en [1]). La ruta solución inicia y termina en un punto o ubicación, y cada ubicación es visitada sólo una vez.

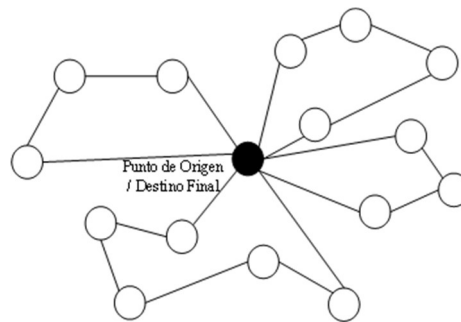


Fig. 2. Ejemplo de Solución para el VRP (basado en [2]). La solución consiste de un conjunto de rutas que inician y terminan en un punto o ubicación, y cada ubicación es visitada sólo una vez y por una sola ruta. Las demandas de estas ubicaciones no deben exceder la capacidad del vehículo que cubre la ruta asignada a éstas.

2. Algoritmos genéticos para problemas de ruteo

Los Algoritmos Genéticos o AG consisten de diversas técnicas de búsqueda que imitan el proceso evolutivo de selección natural o “sobrevivencia del ser más apto”. Dichos algoritmos han sido útiles para encontrar soluciones a diversos problemas combinatorios complejos. Los pasos de un AG se describen en la Tabla 1.

Nótese que la idea de un AG es mejorar soluciones candidatas por medio de la simulación de mecanismos de evolución tales como la selección, cruzamiento y mutación. También, al ser algoritmos de búsqueda local, su desempeño depende de los mecanismos utilizados para explorar el espacio de solución como son los operadores específicos para cruzamiento y mutación [8, 9]. Dentro de las diferentes variantes de estos operadores, el método de cruzamiento es el que más influye en diversificar la estructura de las soluciones.

Aquí es importante señalar que el cruzamiento está sujeto a las representaciones de cromosomas de las soluciones, las cuales pueden ser secuencias binarias, de números enteros, reales o secuenciales [10]. Para los problemas de ruteo en donde las soluciones consisten de secuencias de números enteros únicos (permutaciones), el método de cruzamiento conocido como Cruzamiento de Emparejamiento Parcial o Partially Matched Crossover (PMX) se ha utilizado ampliamente [10,11,12]. Otro aspecto es el

Tabla 1. Pasos generales de un AG.

Paso 0	Inicialización de la población: conjunto inicial de soluciones factibles (individuos) para mejorarse a lo largo del proceso del AG
Paso 1	Evaluación de cada individuo de la población para determinar su “elegibilidad” (por ejemplo, esto puede ser el valor de función objetivo o distancia total de la solución)
Paso 2	Proceso iterativo de búsqueda del AG (generaciones) que se repite hasta que se cumpla la condición de terminación o paro. <ol style="list-style-type: none">Seleccionar padres (pares de soluciones de la población para generar (engendrar) otras soluciones (hijos).Reproducción de padres mediante operadores de cruzamiento y mutación.Evaluar “elegibilidad” de los individuos engendrados (hijos).Actualizar población con los mejores padres e hijos para la siguiente generación.Devolver el mejor individuo y mejores valores en población y promedio para evaluar convergencia.Evaluar si se cumple condición de terminación, de lo contrario, seguir con la próxima iteración (generación).
Paso 3	Regresar mejor individuo en la población final

tamaño de la población, la cual puede restringir o saturar la capacidad de estos operadores para diversificar soluciones. Ambas situaciones pueden dar como resultado “estancamientos” en soluciones locales de baja calidad, aunque el AG se ejecute durante un número extenso de generaciones.

Por ello, es necesario explorar alternativas o configuraciones en los mecanismos del GA para evitar este “estancamiento” en soluciones locales. Como propuesta de este trabajo, se considera el cambio dinámico en el tamaño de la población para crear una interrupción y reinicio de las soluciones candidatas para impulsar la búsqueda en otros espacios de solución y reducir el riesgo de “estancamiento”. Los detalles técnicos del AG en función de los componentes de la Tabla 1 se describen en las siguientes secciones.

3. Algoritmo genético con población variable para el TSP

3.1. Paso 0

Se define inicialmente una población con $P=100$ individuos o soluciones para el TSP en donde el 50% son generadas de manera aleatoria y el 50% son generadas mediante un algoritmo voraz (Greedy Algorithm). Como ejemplo de una solución, se tiene el siguiente considerando 12 ubicaciones:

$$[1 - 9 - 6 - 5 - 10 - 2 - 4 - 8 - 12 - 3 - 11 - 7 - 1].$$

Nótese que esta solución implica una ruta que comienza en la ubicación “1”, procede con la ubicación “9”, y de manera secuencial continua con las ubicaciones “6”, “5”, “10”, “2”, “4”, “8”, “12”, “3”, “11” y “7” para finalmente regresar a la ubicación de origen “1”, la cual en el context logístico y de distribución usualmente corresponde al centro de distribución, almacén, o centro de acopio. Este formato de solución

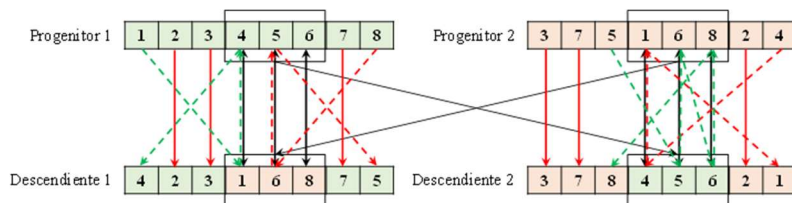


Fig. 3. Ejemplo de cruzamiento PMX (basado en [13]).

corresponde a la representación cromosómica del AG para la cual se deben seleccionar operadores de cruzamiento y mutación apropiados.

3.2. Paso 1

Una vez teniendo la población inicial se procede a evaluar la elegibilidad de cada individuo. En este caso, la elegibilidad depende de la distancia total asociada a la ruta. Para ello, se considera la suma de distancias Euclidianas entre cada ubicación dentro de la ruta.

Dentro del contexto de modelación matemática, la distancia total se puede expresar mediante la siguiente función:

$$C_T = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \tag{1}$$

En donde n es el número total de nodos o ubicaciones a visitar en el problema, c_{ij} es la distancia o costo de viajar al nodo j desde i , (por ejemplo, la distancia Euclídeana entre las ubicaciones i y j) y x_{ij} es la variable binaria de decisión que es igual a “1” si el camino entre i y j se considera dentro de la ruta solución al TSP, y es igual a “0” en caso contrario. Nótese que resolver este problema consiste en encontrar los valores de la variable de decisión x_{ij} que minimicen la función C_T . Esto es, resolver la función objetivo $\min C_T$.

Al tener la población con estos valores de elegibilidad de cada individuo se hace un ordenamiento en base a éste último de manera descendente. Con ello, los individuos (rutas) con menor distancia total se ubicarán al inicio de la población en tanto que aquellos con las mayores distancias (y, por lo tanto, menos elegibles para resolver el TSP) se ubicarán al final de la población.

3.3. Paso 2

- a. **Selección:** Para comenzar la reproducción de individuos es necesario escoger un método de selección apropiado para las soluciones progenitoras. Para este propósito se considera el método de “torneo” en donde los individuos con mejor valor de elegibilidad son candidatos para reproducción. También, de manera aleatoria, una solución se elegirá de la parte inferior de la población para reproducción con una solución de la parte superior.

Tabla 2. Instancias de la base de datos TSPLIB usadas para pruebas.

<i>n</i>	instancia	<i>n</i>	instancia
51	eil51.tsp	198	d198.tsp
52	berlin52.tsp	200	kroA200.tsp
70	st70.tsp	200	kroB200.tsp
76	eil76.tsp	225	ts225.tsp
76	pr76.tsp	225	tsp225.tsp
99	rat99.tsp	226	pr226.tsp
100	kroA100.tsp	262	gil262.tsp
100	kroB100.tsp	264	pr264.tsp
100	kroC100.tsp	280	a280.tsp
100	kroD100.tsp	299	pr299.tsp
100	kroE100.tsp	318	lin318.tsp
100	rd100.tsp	400	rd400.tsp
105	lin105.tsp	417	fl417.tsp
107	pr107.tsp	439	pr439.tsp
124	pr124.tsp	442	pcb442.tsp
127	bier127.tsp	493	d493.tsp
130	ch130.tsp	574	u574.tsp
136	pr136.tsp	575	rat575.tsp
144	pr144.tsp	654	p654.tsp
150	ch150.tsp	657	d657.tsp
150	kroA150.tsp	724	u724.tsp
150	kroB150.tsp	783	rat783.tsp
152	pr152.tsp	1002	pr1002.tsp
159	u159.tsp	1060	u1060.tsp
195	rat195.tsp	1084	vm1084.tsp

- b. Reproducción:** Al tener la selección de pares de individuos para reproducción, la generación de nuevos individuos (soluciones descendientes o hijos) se obtiene mediante los operadores de cruzamiento y mutación.

Como operador de cruzamiento se considera el método PMX, el cual hace una combinación factible de fracciones de cada solución progenitora. Un ejemplo de cómo funciona este operador se presenta en la Figura 3.

En cuanto a los operadores de mutación, se consideran a los métodos de inversión de sub-secuencias entre dos puntos, e intercambio de dos puntos.

- c. Evaluación de elegibilidad:** Una vez obtenidas las soluciones descendientes el valor de elegibilidad de cada una se obtiene mediante la suma acumulada de las **distancias** Euclidianas entre cada ubicación dentro de la ruta equivalente.
- d. Actualización de la población:** Al tener todas las soluciones evaluadas, se hace una unión de las soluciones en la población con aquellas de los descendientes. Esta población total se ordena en orden descendente y se toman las mejores P soluciones para integrar la población para la siguiente generación. Nótese que esto va a ocasionar que, tanto el valor de la mejor solución como el del promedio de las soluciones en la población, se mejoren con respecto a la generación presente.

Inicialmente el AG se ejecuta durante X generaciones con una población fija de P soluciones. Al llegar a este punto se comienza la dinámica de cambio de tamaño de

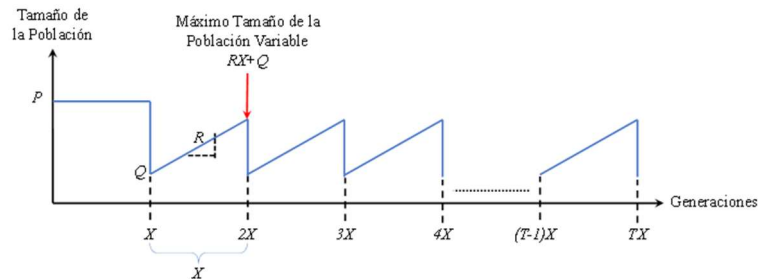


Fig. 4. Dinámica de cambio de tamaño de población para el AG.

población mediante un reinicio a Q soluciones, las cuales se incrementarán a una razón de R unidades por generación. Esta dinámica se ilustra en la Figura 4. Nótese que el proceso de reinicio del tamaño de la población a Q soluciones se realiza cada X generaciones y que, dependiendo de la razón de incremento en R unidades, se puede llegar a tener una población con un máximo de $R \times X + Q$ soluciones que puede ser mayor que la población inicial P .

- e. **Cálculo de convergencia y registro de mejor solución:** Al final de cada generación se hace un registro de la mejor solución, la cual, de acuerdo al esquema de actualización de la población, corresponde a la primera solución en la misma. El valor de elegibilidad de esta solución también se registra para llevar un control de la convergencia del AG.
- f. **Condición de terminación:** Como condición de terminación se define un número máximo de generaciones K . Para este caso, y para propósitos de evaluar el enfoque del cambio de población, se consideran $K=1000$ generaciones.

4. Resultados

El AG fue codificado en OCTAVE y fue ejecutado en una computadora Laptop Intel Core i7-5500U CPU a 2.40 GHz con 8GB RAM. La tasa de cruzamiento por PMX fue del 60% en tanto que la de mutación por intercambio e inversión fue del 30%. Las pruebas se realizaron con las instancias de la base de datos TSPLIB [14]. Los detalles de estas instancias se muestran en la Tabla 2. Como se puede observar, se consideraron 50 instancias en donde 25 instancias tienen de 51 a 195 ubicaciones, y 25 instancias tienen de 198 a 1084 ubicaciones.

Para estas instancias, la Figura 5 muestra el error promedio con respecto a las mejores soluciones. Nótese que se muestra la convergencia del error con el AG con población constante ($P=100$) y con reinicio de la población (población variable con reinicio a $Q = 50$ soluciones). Para este caso se consideró $X = 20$ generaciones y $R = 5$ individuos (para tener una población máxima con $20 \times 5 + 50 = 150$ soluciones).

Como se puede observar, los resultados obtenidos con $P=100$ muestran el mejor desempeño dentro del rango de las 1000 generaciones con un error promedio de 8.5%. El AG con reinicio de la población tiene un error promedio mayor con 9.8%. Si bien, el AG con reinicio se ejecuta más rápido que el AG con población constante, en ambos

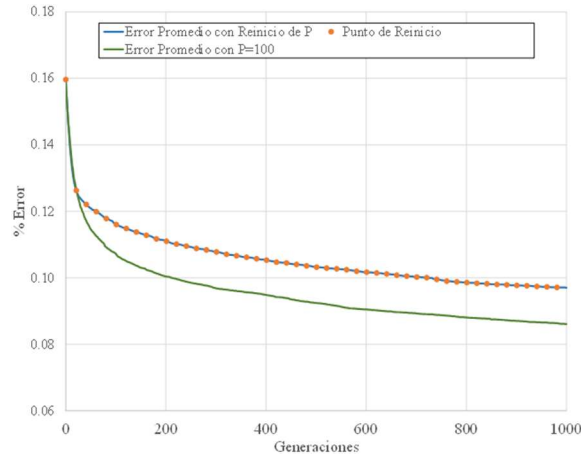


Fig. 5. Convergencia del error promedio con el AG con población fija ($P=100$) y población variable (reinicio de P cada $X=20$ generaciones).

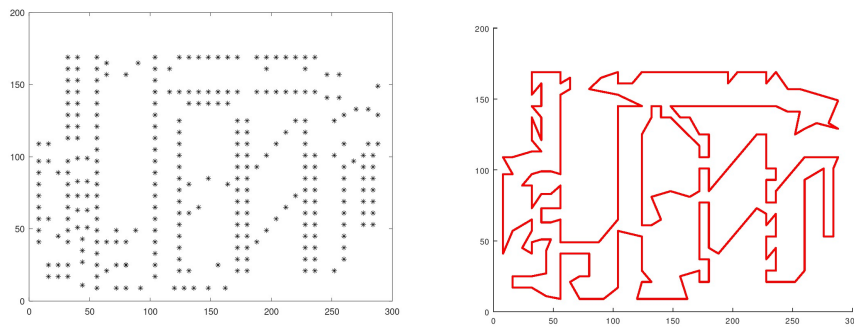


Fig. 6. Solución obtenida para la instancia a280.tsp con el AG con $P=100$ y 2000 generaciones.

casos se observa un decremento en el error, lo cual corresponde con una convergencia progresiva sin estancamiento hacia un óptimo local.

No obstante, se puede observar que la pendiente de convergencia es más pronunciada (o acelerada) en el caso con P constante. Cuando hay reinicio en P , la pendiente tiene una desaceleración a partir de la generación 600, lo cual disminuye la pendiente de la convergencia. Con estos resultados se puede estimar que el cambio en el tamaño de la población puede retrasar la convergencia hacia el óptimo local, lo que implicaría considerar un número mayor de generaciones.

Por otro lado, se verifica que la aplicación de los AG con el operador de PMX pueden brindar soluciones con un error promedio del 8.5% en menos de 1000 generaciones. Al analizar el patrón de convergencia, se puede esperar una disminución mayor del error al extender el número de generaciones. Como ejemplo de esto, la Figura 6 presenta la solución (ruta TSP) final obtenida para la instancia a280.tsp con 2000 generaciones y

$P=100$. Esta solución tiene un error de 6% y presenta un patrón definido característico del TSP.

5. Conclusiones y trabajo a futuro

En el presente trabajo se consideró un AG para resolver redes TSP con más de 200 nodos. En particular, el AG integró el concepto de reinicio de población o población de tamaño variable con el fin de hacer más rápido el AG, y depurar la población para evitar la convergencia hacia un óptimo local de manera prematura.

Si bien el AG con esta integración fue más rápido por el reinicio de población a un tamaño más pequeño, dentro de un rango de 1000 generaciones y 50 instancias de prueba, no se obtuvo una mejora significativa con respecto al AG con tamaño de población fija o constante.

Esta observación es importante para delimitar el ajuste de parámetros de los AG para resolver el TSP. Si bien el tamaño de la población influye en la velocidad y calidad de las soluciones, es más importante mejorar los operadores utilizados para generación de soluciones candidatas (reproducción).

Como trabajo a futuro, se pretende hacer una extensión del operador PMX para hacer el cruzamiento entre varias soluciones [10]. Esto se espera que pueda mejorar la velocidad de convergencia en presencia de poblaciones pequeñas, lo cual contribuiría en el campo de los micro AG (μ AG).

Referencias

1. González-Velarde, J. L., Ríos-Mercado, R. Z.: Investigación de operaciones en acción: Aplicación del TSP en problemas de manufactura y logística. *Ingenierías*, vol. 2, no. 4, pp. 18–23 (1999)
2. Arboleda-Zúñiga, J., Xiomara-López, A., Lorena-Lozano, Y.: El problema de ruteo de vehículos [VRP] y su aplicación en medianas empresas colombianas. *Ingenium*, vol. 10, no. 27, pp. 29–36 (2016)
3. Liu, W., Li, X., Luo, N., Chen, X.: Common grounding optimization for CVRP. In: 8th International Conference on Industrial Electronics and Applications, ICIEA 2013, pp. 1755–1758 (2013) doi: 10.1109/ICIEA.2013.6566652
4. Ho, S. C., Gendreau, M.: Path relinking for the vehicle routing problem. *J Heuristics*, 12, pp. 55–72 (2006)
5. Kumar, V., Panneerselvam, R.: A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator. *International Journal of Computational Intelligence Research*, vol. 13, no. 7, pp. 1717–1733 (2017)
6. Pichpibul, T., Kawtummachai, R.: a heuristic approach based on Clarke-Wright algorithm for open vehicle routing problem. *The Scientific World Journal*, vol. 2013, pp. 1–11 (2013) doi: 10.1155/2013/874349
7. González-Vargas, G., González-Aristizábal, F.: Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio. Parte 1: Formulación del problema. *Revista Ingeniería E Investigación*, vol. 26, no. 3, pp. 149–156 (2006)
8. Méndez, A., Simón, S., Palumbo, D., Chiachera, E., Carnero, M.: Dos Enfoques para la solución del problema de ruteo de vehículos (CVRP): Aplicación a un Caso Real de Recolección de Residuos. *Mecánica Computacional*, pp. 9367–9377(2010)

9. Kora, P., Yadlapalli, P.: Crossover operators in genetic algorithms: A review. *International Journal of Computer Applications*, vol. 162, no. 10, pp. 34–36 (2017)
10. Ting, C. K., Su, C. H., Lee, C. N.: Multi-parent extension of partially mapped crossover for combinatorial optimization problems. *Expert Systems with Applications*, vol. 37, no. 3, pp. 1879–1886 (2010)
11. Deep, K., Mebrahtu, H.: Variant of partially mapped crossover for the travelling salesman problems. *International Journal of Combinatorial Optimization Problems and Informatic*, vol. 3, no. 1, pp. 47–69 (2012)
12. Hussain, A., Muhammad, Y. S., Sajid, M. N.: A simulated study of genetic algorithm with a new crossover operator using traveling salesman problem. *Journal of Mathematics*, vol. 51, no. 5, pp. 61–77 (2019)
13. Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., Dizdarevic, S.: Genetic algorithms for the travelling salesman problem: A Review of representations and operators. *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129-170 (1999) doi:10.1023/a:1006529012972