

INTEGRACIÓN DE ALGORITMOS GENÉTICOS Y REDES DE PETRI COMO PROPUESTA
METODOLÓGICA PARA LA SOLUCIÓN DEL TIME-DEPENDENT TRAVELING
SALESMAN PROBLEM

CESAR DAVID OSORIO CASTAÑEDA

Tesis de maestría
Modalidad de Investigación

Directores de investigación

Juan José Bravo Bastidas, Ph.D
Juan Pablo Orejuela Cabrera, Ph.D



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INGENIERÍA
ÉNFASIS EN INGENIERÍA INDUSTRIAL
SANTIAGO DE CALI, 2021

CONTENIDO

	Pág.
RESUMEN.....	9
1. INTRODUCCIÓN	4
1.1 Antecedentes y motivación	4
1.2 Problema de investigación	7
1.3 Preguntas de investigación y objetivos	8
1.3.1 Pregunta de investigación.....	8
1.3.2 Objetivos de investigación	8
1.4 Relevancia de la investigación	9
1.5 Estructura de la tesis.....	9
2. REDES DE PETRI Y ALGORITMOS GENÉTICOS COMO HERRAMIENTAS PARA PROBLEMAS DE RUTEO	10
2.1 REDES DE PETRI.....	10
2.1.1 Redes de Petri Temporizadas	12
2.1.2 Representación del TSP en RdP.....	13
2.2 ALGORITMOS GENÉTICOS	14
2.3 CONJUNCIÓN DE REDES DE PETRI Y ALGORITMOS GENÉTICOS EN PROBLEMAS COMBINATORIOS	18
3. EL PROBLEMA DE RUTEO TIEMPO DEPENDIENTE.....	19
3.1 OPTIMIZACIÓN COMBINATORIA	19
3.2 TRAVELING SALESMAN PROBLEM (TSP)	20
3.3 TIME DEPENDENT TRAVELING SALESMAN PROBLEM (TDTSP).....	23
3.4 OPTIMAL START TIME PROBLEM.....	34
4. PROPUESTA METODOLÓGICA PARA LA SOLUCIÓN DEL TIME-DEPENDENT TRAVELING SALESMAN PROBLEM.....	37
4.1 Modelamiento propuesto del TDTSP en RdP	38
4.2 Algoritmo genético basado en redes de Petri para la solución del TDTSP considerando el problema de inicio óptimo.....	43
4.3 Validación de la metodología en instancias del problema	52
5. APLICACIÓN A CASO REAL EN PROCESO DE RECOGIDA Y VENTA DE LECHE EN UN MUNICIPIO DE COLOMBIA	57
5.1 Problema abordado.....	58
5.2. Metodología	62
5.3 Resultados de aplicación	65

5.3.1 Caso de estudio.....	66
5.3.2 Análisis y discusiones	68
6. CONCLUSIONES Y TRABAJOS FUTUROS	71
BIBLIOGRAFÍA.....	74

LISTA DE FIGURAS

	Pág.
Figura 1. Tipos de nodos en una RdP.....	10
Figura 2. Nodos y arcos RdP.....	11
Figura 3. Nodos, arcos y pesos de una RdP	11
Figura 4. Regla de disparo de transiciones.....	12
Figura 5. Flujo AG	16
Figura 6. Representación de camino en AG para TSP	17
Figura 7. Soluciones no factibles del AG.....	17
Figura 8. TSP de dos y tres nodos.....	21
Figura 9. Documentos por año del TDTSP en Scopus.....	24
Figura 10. Co-ocurrencia palabras claves en búsqueda del TDTSP	25
Figura 11. Función escalonada del tiempo de viaje	30
Figura 12. El OSTP dentro del TDTSP.....	36
Figura 13. Algoritmo integrador	38
Figura 14. Sub-red.....	39
Figura 15. RdP para el TDTSP.....	40
Figura 16. Dinámica arco (i,j)	41
Figura 17. Ejemplo simulación RdP	42
Figura 18. Primera parte algoritmo simulación.....	43
Figura 19. Segunda parte algoritmo de simulación.....	43
Figura 20. Diagrama de flujo decodificación genes.....	45
Figura 21. Primer avance algoritmo propuesto	46
Figura 22. Algoritmo propuesto "full" optimización tiempo de inicio	47
Figura 23. Convergencia FO instancia de 6 nodos.....	48
Figura 24. Estructura de cromosoma con gen de tiempo	49
Figura 25. Ejemplo de operación de cruce	49
Figura 26. Mutación gen de tiempo	50
Figura 27. Algoritmo propuesto.....	51
Figura 28. Distribución de nodos ciudad de Lyon	52
Figura 29. Resultado óptimo del algoritmo para instancia de 10 nodos	53
Figura 30. Resultado para instancia de 50 nodos	55
Figura 31. Resultado para instancia de 100 nodos	55
Figura 32. Costos del caso.....	60
Figura 33. Costos en función del tiempo de llegada.	61
Figura 34. Penalización por pérdida de ventas.....	61
Figura 35. Pseudocódigo para no optimizados.....	65
Figura 36. Galería municipal de Cumbal	67
Figura 37. Solución caso de estudio.....	69
Figura 38. Comparación soluciones	70

LISTA DE TABLAS

	Pág.
Tabla 1. Explosión combinatoria del TSP.....	22
Tabla 2. Interpretación cromosoma.....	44
Tabla 3. Efecto probabilidad de optimización, instancia de 6 nodos	51
Tabla 4. Efecto probabilidad de optimización, instancia de 10 nodos	54
Tabla 5. Función de tiempo de servicio.	68
Tabla 6. Función de pérdida de ventas	68

RESUMEN

El Time Dependent Traveling Salesman Problem (TDTSP) es una generalización del conocido problema del agente viajero (TSP por sus siglas en ingles), en el que los tiempos de viaje entre un par de nodos depende del instante de tiempo en que dicho viaje se realice. En esta tesis el problema se divide en la exploración de las secuencias de visitas a los nodos, para posteriormente determinar el tiempo de inicio óptimo de dichas rutas. En este sentido se propone un algoritmo híbrido entre algoritmos genéticos y redes de Petri con el que se busca aprovechar, por un lado el buen desempeño de los algoritmos genéticos en problemas combinatorios como el sub-problema de secuenciación de visitas, y por otro lado las ventajas de modelado de las redes de Petri para abordar el sub-problema del manejo temporal del problema. El objetivo fue entonces el diseño del algoritmo híbrido como propuesta metodológica de solución al TDTSP.

Se parte de la caracterización del problema desde los elementos conceptuales introducidos por Fox en 1973, hasta formulaciones más recientes y se propone una representación matemática del problema, así como una representación en red de Petri a partir de la cual se diseña una estructura cromosómica que siempre garantice la factibilidad de todas las soluciones exploradas por el algoritmo genético. Para la fase de evaluación del algoritmo se plantean dos ramas: la primera es la optimización del tiempo de inicio a través de programación matemática, y la segunda es el uso de la lógica de simulación de la red de Petri para obtener la función de desempeño a través del control estricto de la evolución del tiempo que brinda esta herramienta.

El análisis computacional mostró que la rama de la lógica de simulación de la red de Petri tiene menor costo computacional que la optimización del tiempo de inicio, lo cual se traduce en que un mayor uso de la red de Petri en la fase de evaluación del algoritmo, trae mejoras de hasta un 83% en el costo computacional, sin afectar la calidad de las soluciones. La validación en instancias resalta esa calidad de las soluciones obtenidas con el algoritmo propuesto, y la aplicación en el caso real muestra la flexibilidad de la metodología y el potencial de aplicación en diversos contextos. Estos resultados hacen prever este enfoque como promisorio para trabajos futuros.

Palabras clave:

Algoritmos genéticos; redes de Petri; Meta-heurísticas; Time-dependent Traveling Salesman Problem; Tiempo de inicio óptimo.

1. INTRODUCCIÓN

El problema del ruteo de vehículos consiste en minimizar la distancia total recorrida de una flota, garantizando que cada nodo se visite una única vez por vehículo, además de otras restricciones (Hashimoto, Yagiura, & Ibaraki, 2008). Al caso particular en el que se tiene un único vehículo se denomina Traveling Salesman Problem (TSP), el cual es uno de los problemas de optimización combinatoria más famosos, del que se encuentra un amplio desarrollo en la literatura científica (Laporte, 2010).

A pesar de que el TSP goza de un planteamiento relativamente simple, es un problema NP-Hard (Gutin & Punnen, 2003), lo cual ha motivado la investigación hacia algoritmos cada vez más eficientes, y ha dado especial relevancia a los métodos heurísticos y meta-heurísticos de solución, dentro de los cuales se encuentran los reconocidos algoritmos genéticos (AG), cuya aplicación al TSP ha sido frecuente, mostrando buenos desempeños (Chatterjee, Carrera, & Lynch, 1996; Vélez Gallego & Montoya, 2007).

Sin embargo, esa aplicación de los AG para la resolución del TSP, generalmente tiene complicaciones en la factibilidad de los individuos generados, pudiendo obtenerse soluciones que no cumplen las restricciones básicas del problema (Chatterjee et al., 1996), lo cual conlleva a la necesidad de programar dentro de éste, estructuras que restrinjan dicha obtención de soluciones infactibles, o establecer estrategias de penalización o reparación de esos individuos (Mejia, Montoya, Cardona, & Castro, 2012). Agregar dentro del AG una lógica que garantice la factibilidad de las soluciones, significa complejizar el código, disminuyendo probablemente su desempeño (Chatterjee et al., 1996), así como dificultando su aplicación a problemas diversos con restricciones propias, perdiendo así flexibilidad en el método respecto a usos más generales.

Por ejemplo, si se considera una generalización del TSP en el que los tiempos de viaje dependen del momento en que se realiza el recorrido (caso denominado Time-Dependent Traveling Salesman Problem-TDTSP), se hace necesario el control estricto de la evolución del tiempo, requiriendo así, restricciones especiales para ello, lo cual conlleva a una dificultad mayor en la aplicación directa de AG. Y a este respecto, en la presente tesis se estudia el TDTSP y se propone el uso de redes de Petri y algoritmos genéticos como metodología de solución al problema.

1.1 Antecedentes y motivación

El problema del viajero de negocios, o más comúnmente conocido como *traveling salesman problem* (TSP) consiste, coloquialmente hablando, en encontrar la ruta más corta para un agente de negocios que parte de su lugar de origen a visitar un conjunto de ciudades o clientes preestablecidos, para luego retornar de nuevo a su sitio de partida, de tal forma que todas las ciudades o clientes sean visitados una única vez (Gutin & Punnen, 2003). Si se consideran las ciudades como nodos o vértices de un grafo, las conexiones entre las ciudades como las aristas de ese grafo, y las distancias entre ciudades como los pesos de las aristas, entonces, este problema es equivalente al problema matemático de encontrar el ciclo Hamiltoniano de peso mínimo en el grafo.

De forma general, a un problema con un conjunto de soluciones finito o infinito numerable, se le denomina problema combinatorio. Por consiguiente, el TSP es un problema de este tipo, ya que la cantidad de tours posibles entre los nodos es finita. Y si en un problema combinatorio se busca el

objeto o solución que optimice una determinada función objetivo, se habla de un problema de *optimización combinatoria*, en donde el objeto buscado, generalmente, es un número entero o un conjunto de números enteros (como es el caso de los modelos de programación entera), una permutación o una estructura de grafo (Blum & Roli, 2003).

Según Laporte (2010) el TSP es el problema de optimización combinatoria más famoso, cuya acogida se debe, en gran medida, a la naturaleza del problema cuyo planteamiento es sencillo pero su resolución compleja. Dicha complejidad radica fundamentalmente en la generación de la denominada *explosión combinatoria*, que es el fenómeno en el que un pequeño aumento en el tamaño del problema genera un aumento exponencial en su espacio de soluciones, y específicamente para el TSP simétrico de n nodos, el número de ciclos Hamiltonianos posibles es $\frac{(n-1)!}{2}$, lo cual implica que si se tienen 5 nodos la cantidad de soluciones posibles son 12, pero si tan solo se duplican la cantidad de nodos, es decir, se tienen 10, el número de soluciones se eleva a 181440, y ya con 20 nodos la cifra astronómica es de $6,0823E+16$ soluciones. Este hecho hace que la optimización de este tipo de problemas sea computacionalmente costosa y en muchas ocasiones imposible de resolver de manera óptima con los algoritmos conocidos hasta hoy.

Una de las primeras propuestas de solución a este problema fue la de Dantzig, Fulkerson, & Johnson (1954) en la que se plantea un modelo de programación matemática cuya función objetivo optimiza la distancia total recorrida. Posteriormente Miller, Tucker & Zemlin (1960) proponen un mejoramiento al planteamiento de Dantzig et al. (1954), simplificando las restricciones del rompimiento de sub-tours. Estos usos de herramientas de la investigación de operaciones, como la programación lineal entera mixta, generalmente permiten un modelamiento del problema con un alto grado de representatividad de la situación real (Bjorndal et al., 1995). Adicionalmente, la actual disponibilidad de potentes solvers, trae consigo la ventaja de poder resolver óptimamente programas enteros mixtos de un tamaño importante. Sin embargo, para una amplia cantidad de problemas reales, no es posible llegar a la solución óptima en un tiempo razonable (Bjorndal et al., 1995).

Dado que, en general, los problemas combinatorios tienen un conjunto finito de soluciones, siempre existe para ellos un algoritmo exacto de optimización, consistente en la revisión exhaustiva de todo el espacio de soluciones factibles (Duarte, Pantrigo, & Gallego, 2007). Pero esta alternativa es altamente ineficiente debido a la explosión combinatoria característica de este tipo de problemas, que en casos de una cardinalidad muy elevada hace el problema imposible de resolver con las capacidades de cómputo que se tienen en la actualidad. Y es por esto que las heurísticas y meta-heurísticas toman un papel preponderante para este tipo de problemas, ya que son métodos aproximados que permiten encontrar una solución pseudo-óptima de buena calidad en un tiempo razonable (Duarte et al., 2007).

La búsqueda de métodos eficientes de solución ha impulsado mayoritariamente la investigación sobre el TSP, y a hoy existen diversos algoritmos propuestos en la literatura científica (Laporte, 2010). Pero según Chatterjee, Carrera, & Lynch (1996) esta búsqueda ha traído consigo algoritmos que “requieren códigos extremadamente complejos” (p. 490), situación desventajosa respecto a la aplicación de ellos. Mientras que, para estos mismos autores, los algoritmos genéticos son un meta-heurístico “simple y elegante que permite obtener muy buenas soluciones para una variedad de TSP” (p. 491), situándose como uno de los meta-heurísticos más exitosos para el TSP (Vélez-Gallego & Montoya, 2007).

A pesar de ello, la aplicación de algoritmos genéticos al TSP requiere la consideración de estrategias para el tratamiento de las soluciones no factibles que el algoritmo puede producir, como por ejemplo, la generación de individuos que sugieren visitar a un mismo nodo más de una vez, fenómeno que surge especialmente en la operación de cruce (reproducción sexual), y que debe evitarse ya sea

programando restricciones en el algoritmo, estableciendo estrategias de penalización o, como lo proponen Chatterjee et al. (1996), diseñando un algoritmo de reproducción asexual exclusivamente. Sin embargo, esta situación se complejiza mucho más en situaciones reales donde aparecen restricciones particulares adicionales que hacen infactibles a muchas de las soluciones generadas, lo cual es una situación problemática a la hora de buscar generalizar el uso de estos algoritmos en problemas de ruteo. Y a este respecto Van Der Aalst (1996), ha alertado sobre la exclusiva orientación de la investigación científica hacia objetivos de eficiencia computacional, que no necesariamente aporta algoritmos flexibles que sean fácilmente replicables a una amplia generalidad de casos.

Por otro lado, en los últimos 26 años se han registrado esfuerzos en la modelación de problemas combinatorios como el TSP y el scheduling a través de Redes de Petri (RdP), entre los cuales se encuentran trabajos que plantean algoritmos que buscan generalizar la manera en que se transforman o representan estos problemas en estructuras de RdP (Essani & Haider, 2018; Van Der Aalst, 1996). Y sobre ésta idea de modelar los problemas de TSP a través de RdP, Kodama & Nishi (2017) van más allá, proponiendo a las RdP como *meta-modelo*, es decir, como modelo de un modelo previo de programación lineal entera binaria (0,1) planteado para la resolución de un AGVRP (Automated Guided Vehicles Routing Problem), el cual se transforma a una estructura de RdP usando un algoritmo para ello.

Ahora, en una RdP las transiciones generalmente representan la ocurrencia de eventos en el sistema modelado (Huayna, Cortez, & Vega, 2009), entonces si para el problema de ruteo se consideran las visitas a los nodos como los eventos del sistema, se pueden representar dichas visitas como transiciones de la red (Essani & Haider, 2018; Richard, Falcon Chang, Monmarche, & Proust, 1996), de manera que el problema inicial de la optimización de la secuencia de visitas a nodos, se hace equivalente al denominado problema de la secuencia óptima de disparos de transiciones, denotado como OTFSP por sus siglas en inglés (*Optimal Transition Firing Sequence Problem*), que según Kodama & Nishi (2017) consiste en encontrar del conjunto de posibles secuencias de disparos de transiciones, es decir, de aquellas para las que las ecuaciones de estado de la RdP se satisfacen, la que optimiza la función objetivo.

La representación del problema en RdP conduce a una ventaja en virtud de su estructura y su lógica, la cual garantiza que las posibles soluciones al TSP, representadas como secuencias de transiciones, sean siempre factibles, siendo esto un punto a favor frente a los algoritmos genéticos que, como se discutió anteriormente, pueden generar soluciones no factibles. Y esta situación favorable de las RdP es la que Essani & Haider (2018) buscan aprovechar en el marco del TSP múltiple (mTSP), el cual es un problema que en aplicaciones reales con restricciones adicionales “encontrar una solución factible llega a ser un reto” (p. 1), por lo que los autores plantean modelar el problema con una RdP coloreada como estrategia para obtener un modelo que garantice que todas las posibles alternativas sean soluciones factibles al problema y posteriormente proponen experimentos de simulación para buscar una solución mejorada usando el software CPNTools.

En este orden de ideas, las RdP se convierten en una buena alternativa para la representación del problema, sin embargo, en la literatura hay poca presencia de trabajos que usen RdP para el TSP, como se constata en la búsqueda realizada en Scopus el 29 de julio de 2020 de la siguiente ecuación: *TITLE-ABS-KEY("traveling salesman problem") AND TITLE-ABS-KEY("petri nets")*, la cual arrojó 22 registros, de los cuales solamente 5 corresponden a artículos científicos, hecho que permite concluir que las RdP han sido poco utilizadas para el TSP.

Es que a pesar de que las RdP son una herramienta que permite modelar adecuadamente las características de este tipo de problemas, su uso no necesariamente conlleva a una ventaja en cuanto a la solución del problema, ya que “encontrar la solución óptima del OTFSP es computacionalmente complejo” (Kodama & Nishi, 2017. p. 158). Es decir, las RdP conllevan a una representación conveniente, sin que ello aporte a la simplificación computacional en la solución del problema, puesto que el OTFSP es también un problema combinatorio de secuenciación que puede generar espacios de soluciones enormes.

Teniendo en mente lo anterior, si se considera la premisa de que los AG son una herramienta que permite explorar eficientemente el amplio espacio de soluciones generados en los problemas combinatorios, resulta razonable establecer la conveniencia de una estrategia metodológica que busque integrar los AG y las RdP para la resolución del problema, de manera que se combine las ventajas de modelado de las RdP con la capacidad de los AG para la resolución de problemas combinatorios, generando así una sinergia que conlleve a buenos desempeños.

1.2 Problema de investigación

Los antecedentes del apartado anterior sustentan que tanto las RdP como los AG tienen ventajas en su aplicación al TSP. Estos últimos han mostrado buenos desempeños para el problema de la complejidad computacional, mientras que las RdP se han planteado como estrategia de modelación alterna a la tradicionalmente usada de programación lineal entera mixta, respecto a lo cual se han encontrado ventajas como la conversión a un problema de secuencia de transiciones que se puede resolver usando el formalismo matemático que sustenta a las RdP, además de que se hace posible la simulación para aproximarse al mejoramiento de las soluciones.

Adicional a lo anterior, en los antecedentes se identifica también el siguiente par de premisas: la primera consiste en que para el TSP los AG han mostrado buenos desempeños computacionales, pero pueden generar soluciones no factibles; y la segunda es que la representación del TSP en RdP trae como ventaja la obtención del problema equivalente de la secuencia óptima de transiciones en el que siempre se garantiza la factibilidad de las secuencias de visitas, pero el cual es también un problema combinatorio computacionalmente complejo.

Por consiguiente, estas dos herramientas pueden considerarse como potencialmente complementarias, en el sentido de que los problemas de una los resuelve la otra, es decir, el problema de los algoritmos genéticos en la generación de soluciones lo resuelve el modelamiento con RdP que garantiza factibilidad, y el problema de la complejidad computacional en la secuencia óptima de transiciones se resuelve con la eficiencia de los algoritmos genéticos para este tipo de problemas. Así pues, resulta válido plantear la idea de un método integrador entre ambas herramientas que genere sinergias positivas, buscando un buen desempeño conjunto.

Un trabajo muy interesante en esta línea es el de Mejia, Montoya, Cardona, & Castro (2012) en el que se presenta un método que integra RdP y algoritmos genéticos para el problema combinatorio de scheduling en un sistema de manufactura complejo donde el manejo de las restricciones de precedencia de los trabajos en las máquinas, los tiempos de alistamiento dependientes de la secuencia, entre otros, hacen que con el uso exclusivo de algoritmos genéticos los problemas de factibilidad se agudicen. Esta situación los autores la resuelven modelando el problema con una RdP que permite “una representación de los cromosomas fácil de codificar, obtener siempre soluciones factibles y no requerir estrategias de reparación” (p. 802).

En este orden de ideas, cabe preguntarse si existen propuestas que planteen el uso conjunto de ambas herramientas en el marco del ruteo, y para acercarse a una respuesta se realiza una búsqueda en la base de datos referencial Scopus con una ecuación que permita encontrar trabajos que intersecten el TSP con algoritmos genéticos y redes de Petri. La ecuación usada fue: *TITLE-ABS ("traveling salesman problem" OR "TSP") AND TITLE-ABS ("petri nets") AND TITLE-ABS ("genetic algorithms")*, y para el 29 de Julio de 2020 no se encuentra ningún documento relacionado.

También se realizó la búsqueda en Web of Science, usando la siguiente ecuación: *TODOS LOS CAMPOS: (traveling salesman problem) AND TODOS LOS CAMPOS: (genetic algorithms) AND TEMA: (petri nets)*. Dicha búsqueda tampoco arrojó ningún documento, lo cual es una clara evidencia de que en un campo de investigación tan estudiado como es el del TSP, existe un vacío de conocimiento frente al uso conjunto de algoritmos genéticos y RdP para su solución.

Ahora, tanto la complementariedad de los algoritmos genéticos con las redes de Petri para el TSP, como la ausencia de conocimiento a este respecto, justifican una búsqueda de alternativas metodológicas que integren ambas herramientas, primordialmente para resolver el problema de la factibilidad en el genético. Pero si adicionalmente se considera un caso especial del TSP, como en el que se tienen tiempos dependientes (TDTSP), se puede observar otra ventaja del modelamiento con RdP, y ésta es que las transiciones permiten un manejo y seguimiento estricto del tiempo, lo cual es una necesidad intrínseca para el TDTSP, ya que en este tipo de problemas los modelos requieren de variables que controlen el transcurrir del tiempo, haciendo que la modelación matemática se complejice, como se puede observar en el planteamiento realizado en (Malandraki & Daskin, 1992).

Así pues, el problema abordado en ésta tesis es la ausencia de documentación científica frente a alternativas de solución del TDTSP que aprovechen las ventajas brindadas tanto por las RdP, como por los algoritmos genéticos. Concretamente, la investigación consiste en el diseño de un algoritmo que permita aprovechar las ventajas de la integración de ambas herramientas para la solución del TDTSP.

1.3 Preguntas de investigación y objetivos

En el marco del problema de investigación descrito anteriormente, surge la siguiente pregunta de investigación.

1.3.1 Pregunta de investigación

¿Cómo diseñar un algoritmo basado en redes de Petri y algoritmos genéticos como metodología de solución para el TDTSP?

1.3.2 Objetivos de investigación

Objetivo general: Proponer una metodología de solución al traveling salesman problem tiempo dependiente (TDTSP) basada en redes de Petri y algoritmos genéticos.

Objetivos específicos:

- Caracterizar el TDTSP para establecer una representación matemática del problema.
- Representar el TDTSP como red de Petri con el fin de afrontar la complejidad asociada a las condiciones de evolución temporal del problema.
- Diseñar un algoritmo integrado de redes de Petri y algoritmos genéticos para la solución del TDTSP.
- Validar la metodología en distintas instancias del problema.

1.4 Relevancia de la investigación

A lo largo de la presentación anterior de los antecedentes y del problema de investigación, se ha apelado a referenciar las reconocidas fortalezas de las redes de Petri para el modelado de sistemas discretos, y de los algoritmos genéticos para la solución de problemas combinatorios, como premisas que permiten considerar la integración de ambas herramientas como tema de interés para los problemas de ruteo.

Adicionalmente, la ausencia de trabajos respecto a alternativas metodológicas que propongan integrar los algoritmos genéticos y las redes de Petri para la solución del TDTSP, muestra la relevancia de la investigación realizada en esta tesis, la cual contribuye con una propuesta de integración que permite potenciar ambas herramientas desde una complementariedad en la que se use las ventajas de una en pro de las desventajas de la otra, resultando en un único algoritmo de solución para el problema de ruteo tiempo dependiente, que a juicio del autor resulta novedoso dentro del marco de referencia revisado.

1.5 Estructura de la tesis

En este primer capítulo introductorio se argumentan los objetivos y la justificación de la investigación, el segundo capítulo presenta una conceptualización de las redes de Petri y los algoritmos genéticos, así como las maneras tradicionales e independientes de uso en problemas de ruteo, permitiendo con ello identificar las fortalezas y debilidades de cada herramienta que son las premisas con las que se desarrolla la propuesta de la tesis.

En el tercer capítulo se realiza un marco teórico del problema abordado, haciendo especial énfasis en la evolución de las formulaciones matemáticas propuestas en la literatura, las cuales sustentan la representación matemática propuesta en esta tesis. Se finaliza el capítulo discutiendo ideas más recientes como el planteamiento de la descomposición del problema en sub-problemas como el denominado Optimal Start Time Problem.

El cuarto capítulo es donde se desarrolla la propuesta metodológica sustentada en la complementariedad de las redes de Petri y los algoritmos genéticos para la resolución del TDTSP. En este capítulo se representa el TDTSP como red de Petri, se discute la lógica de la simulación de la red y se usa tanto la representación como la simulación como puntos integradores en el algoritmo genético. Luego se valida el algoritmo en instancias de diversos tamaños y se discuten los resultados.

El quinto capítulo es la aplicación en un caso real de la metodología propuesta. Se trata de un proceso de recogida refrigerada de leche en un municipio de Colombia. En el capítulo se plantea y describe el caso, discutiendo las particularidades del problema abordado y se presentan los fundamentos de la implementación del algoritmo desarrollado. Finalmente se resuelve el problema y se analizan los resultados.

2. REDES DE PETRI Y ALGORITMOS GENÉTICOS COMO HERRAMIENTAS PARA PROBLEMAS DE RUTEO

En este capítulo se presenta el sustento teórico de las herramientas de base para la propuesta metodológica que se planteará posteriormente en esta tesis. Para lo cual se parte de las ideas fundamentales sobre las redes de Petri, entendidas como un tipo de grafo dirigido y bipartido, hasta llegar a una extensión en la que se consideran elementos temporales. Luego se describen las representaciones en RdP del TSP encontradas en la literatura. Se sigue con la conceptualización de los denominados algoritmos genéticos y la posterior discusión respecto a las aplicaciones al TSP y la principal problemática que de ello surge. El capítulo finaliza presentando un breve marco de referencia respecto a la conjunción de algoritmos genéticos y redes de Petri para problemas de scheduling en sistemas de manufactura.

2.1 REDES DE PETRI

Las redes de Petri (RdP) son un tipo especial de grafo dirigido y bipartido, propuesto en 1962 por Carl Adam Petri en su tesis doctoral como una potente herramienta que permite representar gráfica y matemáticamente un sistema. Dicha representación matemática se fundamenta en que permite configurar ecuaciones algebraicas que modelan el comportamiento del sistema (Barad, 2003).

A continuación se desarrolla la definición de RdP y algunas características importantes sin buscar ser absolutamente riguroso ni exhaustivo (para un desarrollo teórico riguroso ver (Murata, 1989)). Considérese, entonces, a V como un conjunto de nodos o vértices y a E un conjunto de arcos, a partir de los cuales se define un grafo dirigido o dígrafo G como el conjunto de todos los vértices y arcos, de manera que a un arco e se asocia al par ordenado (v, w) con v y w pertenecientes a V , donde v es el nodo de salida y w es el nodo de entrada. Formalmente:

$$G = (V, E) \text{ tal que } \forall e \in E \text{ se asocia un par ordenado de vertices } (v, w)$$

En particular, una RdP es un grafo dirigido con dos tipos de nodos o vértices (grafo bipartido), denominados plazas y transiciones, las plazas se representan con círculos y las transiciones con barras, como se muestra en la Figura 1. Así pues, se puede definir ahora a P como el conjunto de plazas y a T como el conjunto de transiciones, de manera que $V = P \cup T$ y $P \cap T = \emptyset$.

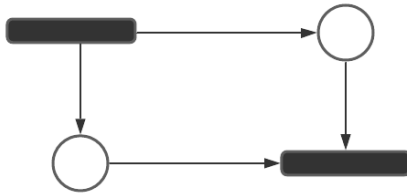
Figura 1. Tipos de nodos en una RdP



Elaboración propia.

Los arcos dirigidos de una RdP únicamente pueden conectar plazas con transiciones, de manera que unos arcos van desde las plazas a las transiciones y otros se dirigen desde las transiciones a las plazas (Huayna et al., 2009), como en el ejemplo mostrado en la Figura 2. De lo anterior se tiene que $P \times T$ es el conjunto de todos los arcos posibles de entrada a las transiciones y $T \times P$ el conjunto de todos los arcos posibles de salida de las transiciones, así que $E \subseteq \{P \times T\} \cup \{T \times P\}$.

Figura 2. Nodos y arcos RdP

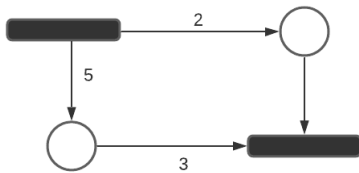


Elaboración propia.

La Figura 2 muestra un modelo gráfico muy sencillo de una RdP, pero cabe anotar que este modelamiento también se puede realizar con elementos matemáticos que permitan representar la misma información que el gráfico, para lo cual se requieren enunciar definiciones pertinentes. Un ejemplo de ello son las definiciones de las funciones de arcos *entrantes* y *salientes* respecto a las transiciones, como se plantea a continuación. Sea I una función cuyo dominio son los pares ordenados generados por el producto cartesiano entre P y T , y de rango binario, que representa las relaciones de entrada de arcos que van de plazas a transiciones, es decir, si para el par (p, t) con $p \in P \wedge t \in T$, $I=1$, entonces existe un arco que sale de p y entra a t , formalmente $I: P \times T \rightarrow \{0,1\}$. A esta función I se le denomina *función de entrada*. Análogamente se define O como la *función de salida*, donde $O: T \times P \rightarrow \{0,1\}$, tal que si $O=1$ para el par (t, p) , entonces existe el arco que sale de t y entra a p .

Ahora, además de ser un grafo bipartido y dirigido, las RdP son un grafo con pesos, es decir, los arcos están etiquetados por un número natural denominado *peso del arco* (ver Figura 3), y por convención, a los arcos con peso unitario *no* se les etiqueta número (Castellanos, 2006). Formalmente se define a W como la función de peso que asocia a cada arco su respectivo número, es decir $W: E \rightarrow \mathbb{N}$.

Figura 3. Nodos, arcos y pesos de una RdP



Elaboración propia.

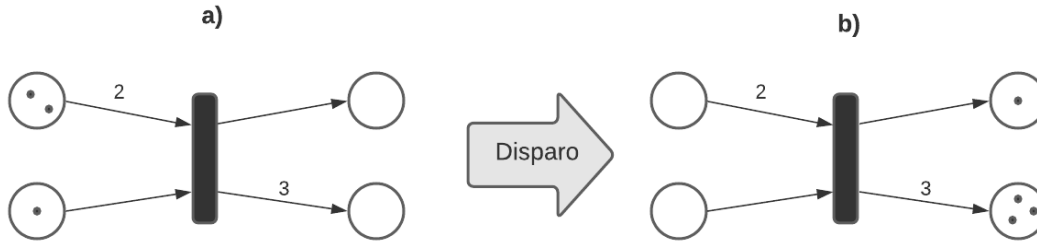
Otro elemento importante de las RdP son los *tokens*, que se representan como puntos negros los cuales residen en las plazas, y la cantidad de tokens que tienen las plazas en cada instante de tiempo define el *estado de la RdP*. A esta cantidad de tokens en cada plaza se le denomina *marcaje* (M), que es un vector cuyos elementos representan la cantidad de tokens en la plaza respectiva, es decir, si la RdP tiene n plazas entonces $M=(m_1, m_2, \dots, m_n)$ donde m_i es la cantidad de tokens en la plaza i (Castellanos, 2006).

La definición de tokens es la que otorga el carácter dinámico de la red, ya que el movimiento de tokens a través de ésta se usa para simular las dinámicas de comportamiento del sistema modelado (Barad, 2003) y dicho movimiento responde a las denominadas reglas de disparo que siguen (Castellanos, 2006), las cuales también se ilustran en la Figura 4:

- Una transición t se dispara cuando cada lugar de entrada p tiene una cantidad de tokens mayor o igual a $W(p, t)$ (Figura 4, a).

- Cuando una transición t se dispara, entonces ésta consume tokens de sus plazas de entrada, y produce tokens a sus plazas de salida, tantos como los pesos de sus respectivos arcos lo indiquen (Figura 4, b). Así que el vector de marcado M sufre un cambio después de que se dispara la transición t , es decir, $M \xrightarrow{t} M'$, de manera que $M'(p) = M(p) - I(p, t) * W(p, t) + O(t, p) * W(t, p); \forall p \in P$, donde $M'(p)$ es a la componente de M' correspondiente a la plaza p .

Figura 4. Regla de disparo de transiciones



Elaboración propia

2.1.1 Redes de Petri Temporizadas

Las RdP han tenido muchas extensiones (Barad, 2003), dentro de las que se incluyen las denominadas *redes de Petri temporizadas*, que tienen la característica de incluir elementos temporales en sus dinámicas, para lo cual se clasifican las plazas en dos tipos: las plazas denominadas “operacionales” a las cuales se les asocia un tiempo y se denotan como Op , y las plazas no temporizadas denotadas como R . De manera que $P - Op = R$ y $P = Op \cup R$. Intuitivamente, una plaza operacional o temporizada es aquella en la que un token debe permanecer un determinado lapso de tiempo antes de que se active la transición de salida.

Este nuevo elemento de la red, que es el tiempo, se formaliza con la definición del *vector de tiempos* denotado como τ , cuya dimensión es equivalente a la cantidad de plazas de la red, y cada componente representa, entonces, el tiempo asociado a la plaza respectiva. Con esto en mente, se puede definir la RdP temporizada como una 6-tupla $N = (P, T, E, W, M_0, \tau)$, donde M_0 es el vector de marcaje inicial.

En este tipo de redes las reglas de disparo requieren tener en consideración los tiempos de operación en las plazas, ya que si una transición tiene una plaza de entrada temporizada, entonces el disparo de la transición ocurre después de que el token haya estado en la plaza ese tiempo de operación definido para ella. Así que cuando un token llega a una plaza temporizada, es necesario controlar el tiempo que se está en dicha plaza o equivalentemente, el tiempo que resta de proceso, ya que cuando ese tiempo restante sea nulo es cuando se dispara la transición relacionada. Para lo cual, se define entonces el vector de *tiempo restante de proceso* $[M_r]$ como la estructura algebraica que contiene en su i -ésima posición el tiempo restante para el disparo de las transiciones de salida de la plaza i . Naturalmente que la dimensión del vector M_r debe coincidir con la dimensión del vector P . Y M_r se rescribe cada que una transición se dispara. También cabe anotar que las plazas temporizadas deben restringirse de aceptar más de un token para evitar duplicar el tiempo restante de proceso de una única plaza (Mejía & Odrey, 2005).

2.1.2 Representación del TSP en RdP

Respecto a la aplicación de las RdP como herramienta de modelado, estas gozan de aceptación entre la comunidad científica para la representación de sistemas paralelos, asíncronos, concurrentes y estocásticos (Castellanos, 2006; Murata, 1989), lo que se ajusta muy bien a casos industriales reales como lo pueden ser los sistemas de manufactura flexible, y los problemas de scheduling (Richard et al., 1996). Lo cual se ve potenciado aún más con la inclusión sobre esas RdP básicas, de extensiones como color y tiempo, ya que estos son desarrollos orientados a dotar la herramienta de un grado de aplicación amplio, facilitando mayormente el modelado de sistemas industriales complejos (Van Der Aalst, 1996). Pero a pesar de las mencionadas ventajas de las RdP en el modelado de problemas de scheduling, y de las más de 4 décadas de amplia investigación sobre dicho problema, las publicaciones de trabajo conjunto entre ambos tópicos son escasas (Van Der Aalst, 1996).

Entendiendo el concepto de scheduling como el problema que busca la secuencia óptima de trabajos sobre recursos escasos, en un determinado lapso de tiempo (Van Der Aalst, 1996); fácilmente se puede interpretar como un problema equivalente al ruteo. Y pensando específicamente en el TSP se tiene la correspondencia directa entre la secuencia óptima de trabajos con la secuencia de clientes a visitar, sobre un único recurso que es el vehículo. En el apartado 1.1 del presente trabajo se encontró que para el TSP también resultan escasas las publicaciones que trabajen el modelado a través de RdP.

Aunque son escasos los artículos que planteen el uso de las RdP para problemas de secuenciación como el scheduling y el ruteo, esta idea se empezó a vislumbrar desde 1996 con el trabajo de Van Der Aalst que se enfoca específicamente al scheduling, planteando el modelado a través de RdP bajo la convención de que las transiciones representen la ocurrencia de eventos y las plazas representen condiciones, como es comúnmente usado en la literatura (Castellanos, 2006), de manera que en el modelo de Van Der Aalst las transiciones representan los eventos de inicio y finalización de trabajos y las plazas los recursos o máquinas para dichos trabajos, lo cual tiene la importante ventaja de permitir la transformación del problema de secuenciación de trabajos al problema de *secuenciación de transiciones*, ya que las operaciones son representadas por estas. Ventaja que radica en que las *secuencias de transiciones* generadas tienen la garantía de ser siempre soluciones factibles, lo cual es importante especialmente para problemas complejos en los que la búsqueda de soluciones factibles es ya un reto.

Respecto al problema análogo de ruteo, Essani & Haider (2018) proponen un algoritmo general para representar problemas de TSP múltiple (problemas en los que se involucran varios viajeros) en una red de Petri coloreada; como estrategia para llegar al problema de secuencia de transiciones que garantiza factibilidad en las alternativas de solución. Pero, al igual que Van Der Aalst (1996), mencionan que el problema siguiente es encontrar la secuencia óptima de transiciones, para lo cual usan la estrategia de simular de la red a través del software CPNTools para buscar una buena solución al problema.

De estos trabajos mencionados se puede observar que la principal ventaja de la representación del TSP en RdP, consistente en obtener el problema equivalente de secuencias de transiciones, tiene la limitante de que la optimización de este problema también es computacionalmente complejo (Kodama & Nishi, 2017), lo cual hace necesario una complementación con estrategias de solución, las cuales parten de la garantía de factibilidad de las alternativas evaluadas. Por ejemplo, en Essani & Haider (2018) la estrategia de solución fue la experimentación simulada, mientras que en Kodama & Nishi (2017) la propuesta radica en la conversión de un modelo programación lineal entera binaria para el TSP, en una RdP, con el objetivo de llegar al problema de la secuencia óptima de transiciones,

para generar expresiones matemáticas derivadas de las ecuaciones de estado de la red y modificar el modelo matemático inicial involucrando estas expresiones válidas demostrando que los tiempos de solución mejoran significativamente.

Una estrategia similar en el uso de modelos matemáticos para la secuencia de transiciones fue planteada por Richard et al (1996) en la que se modela el TSP a través de una RdP temporizada con el objetivo de generar un modelo de programación lineal entera asociado a la red, que optimice la secuencia de transiciones, generando así la ruta óptima del TSP. Aunque cabe anotar que el modelo de programación lineal entera planteado es similar al MTZ ((1)-(4)), con la diferencia de que la restricción para el rompimiento de sub-tours cambia, pero este cambio es solo validado para instancias de 6 clientes o menos, lo que limita su aplicación a problemas reales.

Otra estrategia interesante planteada en la literatura es la realizada por Capkovic (2012) en la que se modela un TSP a través de RdP donde las transiciones representan el depósito y los puntos a visitar, y las plazas corresponden a las aristas o los caminos que conectan los nodos, esto para obtener una representación que permita el uso de métodos matemáticos de la teoría de los ciclos Hamiltonianos. Sin embargo, esta estrategia sufre también la complejidad computacional generada por dimensiones crecientes de los problemas a resolver.

En este apartado se pueden observar los intentos publicados en la literatura respecto al desarrollo de metodologías que permitan usar eficientemente las ventajas del modelado en RdP sobre problemas combinatorios como el TSP, y se puede juzgar que la investigación en este sentido es escasa, en donde aún no se evidencian trabajos que busquen adoptar herramientas meta-heurísticas para tal propósito.

2.2 ALGORITMOS GENÉTICOS

Según Soto (2007) existen dos elementos importantes en un problema de optimización, el primero es el diseño de un modelo que abstraiga la realidad con un alto grado de representatividad, y una vez se tiene un modelo adecuado, la segunda cuestión es cómo resolverlo, es decir, elegir el método de solución. Particularmente en los problemas complejos para los que no se conoce un algoritmo exacto de tiempo polinomial (como lo es el TSP), generalmente se requiere el uso de *algoritmos aproximados* que permitan resolver problemas reales para los que los espacios de soluciones son de una cardinalidad elevada, y que por tanto, hacen inviable el tiempo requerido por métodos exactos (Duarte et al., 2007)

Según Halim & Ismail (2019) los algoritmos aproximados se pueden clasificar en dos tipos: los algoritmos heurísticos y los meta-heurísticos. El término heurística proviene del griego *heuriskein* (εὕρισκειν), verbo que significa “encontrar” (Blum & Roli, 2003), y los métodos heurísticos se entienden como procedimientos o técnicas no rigurosas basadas comúnmente en reglas empíricas, que buscan solucionar problemas difíciles de modo sencillo y rápido (Duarte et al., 2007). Siguiendo a estos mismos autores, el principal problema de los algoritmos heurísticos radica en la común ausencia de mecanismos que les permitan la exploración del espacio de soluciones más allá de los óptimos locales, lo cual condena a estos métodos a quedar atrapados en dichos óptimos locales.

Los meta-heurísticos surgen como respuesta a este problema, ya que, como su nombre lo indica con el prefijo *meta*, van más allá o están en un nivel superior a las heurísticas, sugiriendo con esto un marco de alto nivel que se integra a las heurísticas básicas con el fin de dirigir efectivamente la exploración del espacio de soluciones, trascendiendo de la optimalidad local. El término *meta*-

heurística se atribuye al trabajo de Glover (1986) en el que se mencionan como herramientas de búsqueda a través de movimientos en el espacio de soluciones guiados generalmente por estrategias de aprendizaje en las que confluyen los campos de la inteligencia artificial y la investigación de operaciones.

Dentro de las meta-heurísticas más conocidas se encuentran la colonia de hormigas (ACO), la búsqueda tabú (TS), el recocido simulado (SA) y los algoritmos genéticos (GA). Estos últimos se clasifican dentro de las denominadas *meta-heurísticas poblacionales* las cuales deben su nombre al hecho de que las iteraciones del algoritmo están en función de un conjunto de soluciones llamado población, por lo que su eficiencia depende fuertemente de las estrategias de manipulación de esta (Duarte et al., 2007)

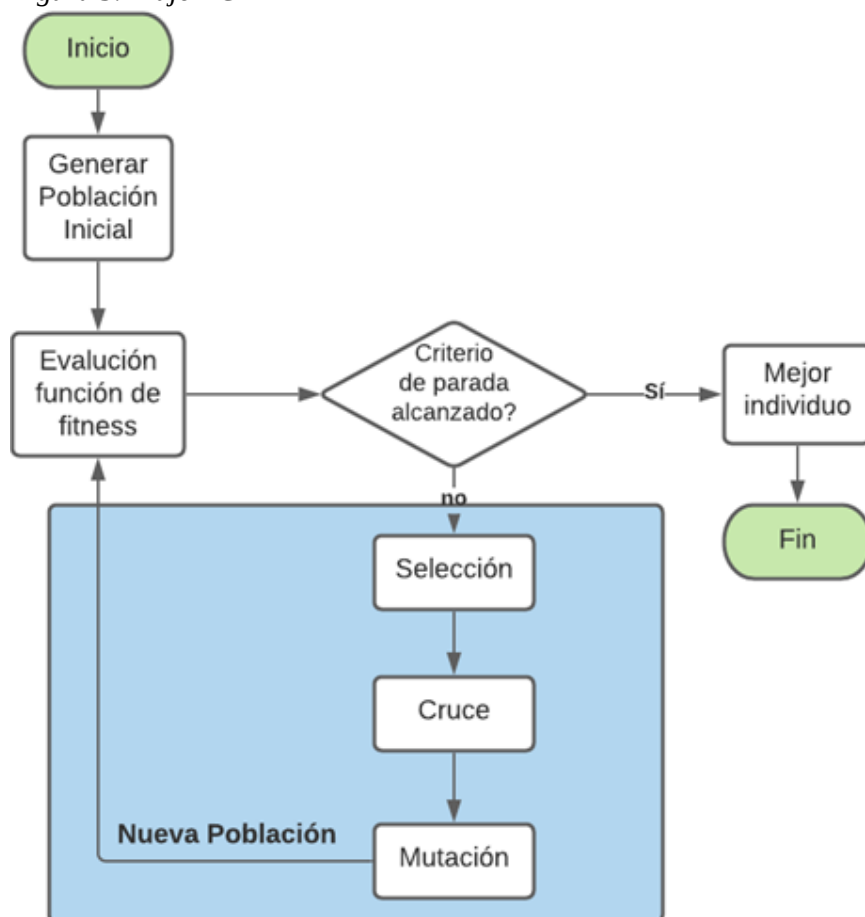
Fue Holland (1975) quien introdujo la estructura básica de los algoritmos genéticos, los cuales se inspiran en la idea de la evolución darwiniana basada en la consideración de que los individuos de una población que mejor se adaptan al ambiente, tienen mayores probabilidades de vivir más tiempo, lo que aumenta a su vez las probabilidades de generar descendencia que herede sus buenas características. Mientras que para los individuos con peor adaptación escasean las probabilidades de supervivencia, por lo que acaban extinguiéndose (Duarte et al., 2007)

Este mecanismo de selección natural basado en la evolución de una población, se emula algorítmicamente usando operaciones de selección, cruce y mutación, como se muestra en la Figura 5, y cuya descripción se presenta a continuación con base en Kamrani & Gonzalez (2003). El algoritmo busca evolucionar una población, entendiendo por ella como un conjunto de individuos denominados *cromosomas*, los cuales representan soluciones al problema. Esta codificación, denominada genotipo, comúnmente se realiza con una cadena de símbolos que pueden ser números o letras, y por ello es que en la aplicación de algoritmos genéticos se debe tener en cuenta que cada problema es diferente, y por tanto la codificación de las soluciones en cromosomas debe representar acertadamente lo que sea que signifique una solución del problema, para dotar de sentido al algoritmo.

Se empieza pues, con una *población inicial* predeterminada, comúnmente generada aleatoriamente, a partir de la cual se inicia el proceso de evolución iterativo del algoritmo. Cada iteración produce una *generación* nueva con nuevos cromosomas que son *descendencia* de la anterior, y consiste en evaluar a los individuos de la población usando la función de fitness, también denominada función de adaptación, a partir de la cual se definirán los mejores individuos para la evolución, con el objetivo de que esos mejores cromosomas tengan mayores probabilidades de sobrevivir o permanecer más tiempo en las generaciones, y de acuerdo a Holland (1975), a medida que transcurren las generaciones, el algoritmo debe converger al mejor cromosoma, y en el mejor de los casos este podría representar la solución óptima del problema, aunque esa optimalidad no se garantiza.

En la Figura 5 se observa el diagrama de flujo detallado del algoritmo que inicia con la generación aleatoria de N individuos de la población inicial, a los que se les procede a evaluar la función de fitness. De acuerdo con el desempeño de esa función se realiza la selección de los padres para la próxima generación, padres sobre los cuales se hace la operación de cruce que genera los hijos. Luego, de acuerdo con la probabilidad de mutación, se mutan algunos de los individuos para finalmente obtener la *nueva generación* a la cual se le vuelve a realizar la iteración hasta alcanzar el criterio de parada después del cual se espera obtener el mejor individuo.

Figura 5. Flujo AG



Fuente: diapositivas de clase profesor Juan Pablo Orejuela Cabrera

Algoritmos Genéticos para el TSP

Fue en 1985 con los trabajos de Brady (1985) y Grefenstette et al., (1985) que se iniciaron las aplicaciones de los algoritmos genéticos para resolver el TSP, y desde entonces muchos investigadores han venido planteando algoritmos mejorados para aumentar el desempeño en la resolución del problema (Sun, 2020). Ese amplio uso de los algoritmos genéticos para el TSP, según Vahdati, Yaghoubi, Poostchi, & Naghibi S. (2009), se debe principalmente a los cortos tiempos requeridos en la generación de buenas soluciones. Y respecto a estas ventajas de los algoritmos genéticos Chatterjee et al. (1996) los califica como un “heurístico simple y elegante que permite obtener muy buenas soluciones” (p. 491)

Como se mencionó anteriormente, cuando se piensa aplicar un algoritmo genético es fundamental definir adecuadamente la codificación de los cromosomas, de manera que estos representen correctamente las soluciones del problema. Según Chatterjee, Carrera, & Lynch (1996) de esa adecuada representación depende el éxito del algoritmo. En muchas aplicaciones se utilizan cadenas binarias como codificación de cromosomas, pero para el caso específico del TSP ese tipo de representación no es una aplicación efectiva, ya que no permite asociar directamente una solución con un cromosoma, es por eso que la representación comúnmente usada en la literatura consiste en

definir los cromosomas como permutaciones de nodos a visitar en lo que se denomina *representación de camino*, es decir, los cromosomas son una secuencia de elementos (regularmente números naturales asociados a nodos específicos) cuyo significado se puede interpretar de la siguiente manera: si una ciudad i está ubicada en la posición j del cromosoma, entonces la ciudad i es la j -ésima ciudad visitada en el tour que representa el cromosoma (Larrañaga, Kuijpers, Murga, Inza, & Dizdarevic, 1999; Vahdati et al., 2009). En la Figura 6 se ilustra un ejemplo de este tipo común de representación para el caso de 4 nodos, en donde a cada nodo se le asocia un número del 1 al 4, y por tanto, el cromosoma mostrado se lee secuencialmente de izquierda a derecha como los nodos a visitar, generando así una posible solución al problema o tour.

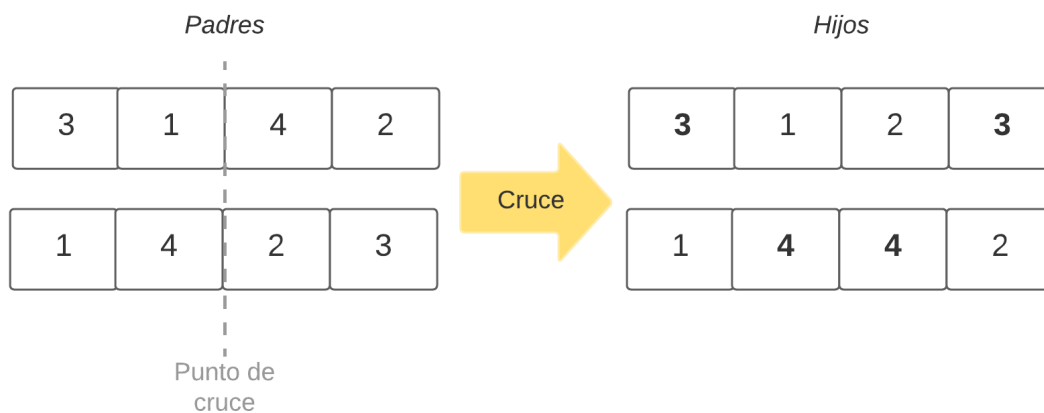
Figura 6. Representación de camino en AG para TSP



Elaboración propia.

Sin embargo, esta representación estándar en los algoritmos genéticos para el TSP tiene el problema de que la generación de nuevos individuos no garantiza factibilidad en las soluciones (Larrañaga et al., 1999), especialmente en las operaciones de cruce como se menciona en (Chatterjee et al., 1996), en las que se pueden generar secuencias no factibles en donde se repiten visitas a nodos como se observa en la Figura 7 para el ejemplo de 4 nodos y un punto de cruce arbitrario asignado al segundo gen. Este tipo de situaciones conlleva a que se requiera agregar en el algoritmo estrategias de reparación de soluciones (Mejia et al., 2012) y se puede complicar en la medida que el problema contenga restricciones particulares como por ejemplo las condiciones de precedencia, y todo ello implica una complejización del código del algoritmo.

Figura 7. Soluciones no factibles del AG



Elaboración propia.

Precisamente esta problemática en la aplicación de los AG al TSP, es una de las cosas que se busca resolver con el algoritmo propuesto en esta tesis, que integra redes de Petri en el AG para aprovechar las ventajas del modelado en RdP entre las que se encuentra la garantía de factibilidad de las soluciones exploradas.

2.3 CONJUNCIÓN DE REDES DE PETRI Y ALGORITMOS GENÉTICOS EN PROBLEMAS COMBINATORIOS

Dentro de los problemas típicos de optimización combinatoria se encuentran el problema del viajante de negocios (Traveling Salesman Problem-TSP), el problema de asignación cuadrática (Quadratic Assignment Problem-QAP), los problemas de planificación (Scheduling Problems), los problemas de cortes sobre grafos (Cut Set Problems), entre otros (Duarte et al., 2007). Generalmente los problemas de scheduling se enmarcan dentro sistemas de manufactura cuyas características generan restricciones propias que requieren ser consideradas en la representación del problema. Por ejemplo, son comunes restricciones como las relaciones de precedencia de los trabajos en las máquinas, los tiempos de alistamiento dependientes de la secuencia, estaciones con máquinas en paralelo no necesariamente idénticas y capacidad limitada de almacenamiento temporal en las estaciones.

Para este tipo de sistemas es relevante contar con una herramienta de modelado que describa apropiadamente las características propias del sistema, y en este sentido, J. Caballero & Mejía (2006) presentan a las RdP como una herramienta adecuada en el modelado de sistemas discretos como lo son los sistemas de manufactura flexible (SMF), y mencionan el uso que algunos autores han hecho de las RdP para modelar este tipo de sistemas. Sin embargo, a pesar de las ventajas de modelado de las RdP, su uso en el marco de los SMF ha sido escaso, debido a que dicha representación no disminuye la complejidad asociada a la solución del problema de scheduling del sistema, que se representa a través de posibles estados de la red cuya exploración resulta ser un problema combinatorio con un cantidad de posibles soluciones que puede ser enorme.

Según J. Caballero & Mejía (2006), los autores que han empleado las RdP para modelar un SMF, utilizan la red únicamente para controlar la evolución del sistema, pero la programación de los trabajos en éste (scheduling) se realiza de manera independiente a través de la aplicación de algún método heurístico. Partiendo de este punto, los autores antes citados, proponen un nuevo enfoque para la programación de la producción de un SMF, en el que se representa el sistema a través de RdP, y con éste modelo se resuelve el problema de scheduling usando algoritmos genéticos que son un meta-heurístico exitoso para este tipo de problemas combinatorios que permiten explorar eficientemente el espacio de soluciones.

Este nuevo enfoque permite, entonces, combinar las ventajas de modelado de las RdP con la capacidad de los AG para resolver problemas combinatorios. El amplio reconocimiento de dichas fortalezas tanto de las RdP como de los AG permiten considerar su integración como tema de interés en la programación de la producción (J. Caballero & Mejía, 2006). Y sobre esta línea de investigación se han desarrollado trabajos posteriores como (J. P. Caballero, Mejía, & García, 2013; Mejia et al., 2012). Sin embargo, esta propuesta integradora se ha limitado únicamente a problemas de scheduling en el marco de sistemas de manufactura, lo cual es un antecedente importante que hace prever dicho enfoque como promisorio para otros tipos de problemas combinatorios como lo son los problemas de ruteo.

3. EL PROBLEMA DE RUTEO TIEMPO DEPENDIENTE

Este capítulo es un recorrido por el marco teórico del problema que se busca resolver con la estrategia planteada y estudiada en esta tesis. Se empieza desde el ámbito general de la optimización combinatoria hasta ubicarse en el problema particular del TSP, con el que se asientan las bases para el punto central del capítulo que consiste en el estudio de la generalización del problema en que se consideran tiempos dependientes (TDTSP), haciendo especial énfasis en la caracterización y las formulaciones matemáticas de éste, a partir de las cuales se plantea el modelo matemático propuesto en esta tesis. Finalmente, se hace hincapié en el sub-problema de la determinación del tiempo de inicio óptimo dada una secuencia, que está inmerso en el problema completo tiempo dependiente, y se discute el potencial uso del modelo matemático de dicho sub-problema dentro de algoritmos heurísticos o meta-heurísticos para la solución del problema completo.

3.1 OPTIMIZACIÓN COMBINATORIA

En términos matemáticos, un *problema de optimización* P tiene la siguiente forma:

$$(P) \text{ Optimizar: } f(x)$$

$$x \in F \subseteq SS$$

Donde x es el vector de variables, $f: F \rightarrow \mathbb{R}$ es la función a optimizar, F es el conjunto de soluciones factibles y SS es el espacio de soluciones. Entonces el conjunto F se puede representar como:

$$F = \{x \in SS / g_i(x) \leq 0, i = 1, 2 \dots m\}$$

De manera que las expresiones $g_i(x) \leq 0, i = 1, \dots, m$ modelan las restricciones del problema, y a un $x \in F$, se le denomina *solución factible* de P. Pero si además de ser factible (y considerando un problema de minimización), una solución (x) cumple que

$$f(x) \leq f(y), \forall y \in F$$

Se dice que x es ***solución óptima***.

Esto se traduce en buscar la mejor configuración de las componentes de x respecto de f , de lo que surgen naturalmente dos categorías (Blum & Roli, 2003): los problemas en los que las componentes de x pertenecen a $\mathbb{R}$ y aquellos en los que las componentes de x pertenecen a $\mathbb{Z}$, es decir, problemas de variables continuas o de variables discretas, respectivamente.

Ahora, un *problema combinatorio* se define como la búsqueda de un objeto dentro de un conjunto finito, o infinito numerable, de posibilidades (Duarte et al., 2007) y si se piensan estos objetos como números enteros, entonces se tiene que dentro de los problemas de optimización discretos se encuentran los denominados *problemas de optimización combinatoria* (Blum & Roli, 2003).

Así pues, la optimización combinatoria estudia problemas como P en los que SS es un conjunto discreto, es decir, en los que se tiene como objetivo encontrar el valor óptimo de una función cuyo dominio es un conjunto numerable (Bjorndal et al., 1995) (para una definición con mayor grado de formalidad del problema general de optimización combinatoria ver (Blum & Roli, 2003)).

Para la gran mayoría de problemas de optimización combinatoria se tiene que el conjunto de soluciones factibles (F) es finito (Duarte et al., 2007), caso en el que la exploración exhaustiva (enumeración) del conjunto F genera una solución exacta; la cual es una opción inviable para muchos de los problemas reales en la práctica, ya que la cantidad de posibles soluciones suele ser extremadamente alta (Bjorndal et al., 1995). Y esta elevada cardinalidad del espacio de soluciones, característica en problemas combinatorios, conlleva al reconocimiento general de estos problemas como del tipo NP, es decir, aquellos para los que actualmente no se conoce un algoritmo de complejidad polinómica que resuelva el problema de manera exacta (Bjorndal et al., 1995; Duarte et al., 2007; Gutin & Punnen, 2003).

Dentro de la optimización combinatoria se encuentran reconocidos problemas como el problema de la asignación cuadrática, problemas de cortes sobre grafos, de programación de producción (scheduling) y el problema del viajero de negocio (TSP), entre otros (para una descripción de estos problemas revisar (Duarte et al., 2007)). Y en este marco general, la presente investigación se concentra en el problema de ruteo TSP, el cual se introduce a continuación.

3.2 TRAVELING SALESMAN PROBLEM (TSP)

El problema del viajero de negocios o TSP busca el tour de menor distancia total recorrida por un agente de negocios que parte desde su lugar de origen hacia unos clientes predefinidos, de manera que visita a cada cliente una única vez, y después de que los haya visitado a todos regresa a su punto de partida.

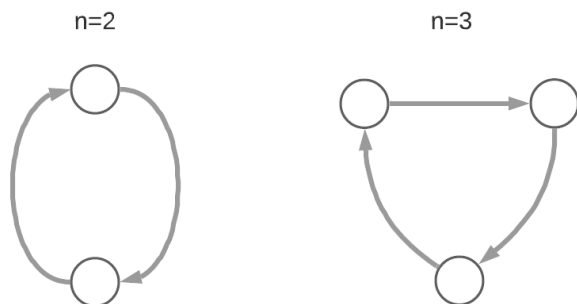
Este problema es análogo al juego planteado por el matemático William Rowan Hamilton en 1856 consistente en un dodecaedro (polígono de doce caras) en cuyos vértices se incrustan alfileres, de manera que se reta a enrollar con una cuerda cada alfiler, pasando por las aristas del polígono de tal forma que se agoten todos los alfileres sin repetición y que se vuelva al alfiler inicial, es decir, se trata de un recorrido cerrado, en el que el único alfiler al que se le permite ser enrollado dos veces es al de partida y finalización.

La naturaleza geométrica del juego de Hamilton permite una intuitiva abstracción del problema desde la teoría de grafos, ya que se tiene una clara correspondencia entre los vértices del polígono con los del grafo, y lo mismo ocurre con las aristas. Esta representación es muy usada en el estudio del TSP.

Siguiendo a Laporte (2010), si en el TSP se considera la casa del agente de negocios, así como sus clientes, como vértices o nodos, y los posibles viajes entre nodos como las aristas, entonces el problema se puede representar como un grafo completo *no* dirigido $G = (V, E)$, donde $V = \{1, 2, \dots, n\}$ es el conjunto de nodos y $E = \{(i, j) / i, j \in V, i \neq j\}$ el conjunto de aristas. Sobre E se define la función de pesos o de costo $C: (i, j) \rightarrow \mathbb{R}$, que asigna a cada arista (i, j) su respectivo peso; esta función de costos, para el TSP, se interpreta como una función de distancias, y regularmente se representa como una matriz C_{ij} en cuya componente (i, j) se encuentra la distancia recorrida del viaje desde i hacia j (Gutin & Punnen, 2003). En este punto cabe anotar que la anterior descripción corresponde al caso del *TSP simétrico*, que es la denominación dada al problema en el que la distancia recorrida por ir desde i hacia j , es equivalente a la distancia del recorrido desde j hacia i , esto $\forall i, j \in V$. Pero si se tiene el caso del *TSP asimétrico*, que es aquel en el que la distancia de i hacia j es distinta de la de j hacia i , entonces la representación del problema es ahora un grafo completo *dirigido* en donde el conjunto E no corresponde a aristas sino a arcos y en la función de costos los pares (i, j) son ordenados.

Es claro que para el problema simétrico de dos o tres nodos la respuesta es trivial, ya que solo hay un posible tour, como se muestra en la Figura 8 en la que se ilustra la única opción para ambos casos, ya que por la simetría, el tour en el sentido de las manecillas del reloj es equivalente al caso del tour en contra de las manecillas del reloj, puesto que las distancias entre cada par de nodos son equivalentes para ambos sentidos del viaje.

Figura 8. TSP de dos y tres nodos



Elaboración propia.

Pero para los casos de cuatro o más nodos, existen diversas soluciones al problema, de entre las cuales se desea encontrar la de menor distancia total recorrida, y un dato pertinente para este propósito es el de la cantidad de dichas posibilidades. Para entender el conteo de esta cantidad de soluciones factibles piénsese en una situación de n nodos, incluyendo el nodo inicial o depósito. En el primer instante se está precisamente en ese nodo de partida, así que parado allí, hay $n-1$ posibles nodos para visitar. Ahora, suponiendo que se escoge alguno de esos $n-1$ nodos posibles y se viaja hasta él, entonces una vez parado en ese segundo nodo se sabe que no se puede volver al anterior, ni tampoco se puede ir a él mismo, por lo que en ese instante la cantidad de posibles alternativas para continuar con un tour factible son los $n-2$ nodos por los que aún no se ha pasado, y así sucesivamente hasta que se agoten todos los nodos, de manera que la única posibilidad del último nodo es volver al depósito.

Por consiguiente la cantidad total de soluciones es el producto entre las posibilidades en cada instante, partiendo de las primeras $n-1$ alternativas del instante inicial, hasta la única alternativa del último nodo visitado que es volver al depósito, lo cual es una estructura factorial de la forma $(n-1)!$. Sin embargo, de ese total de posibilidades para el caso simétrico, se tiene la equivalencia entre cada par de rutas con la misma secuencia de nodos visitados pero en sentidos opuestos, es decir, si $n=4$ y considerando el depósito como el nodo 1, entonces el tour 1-2-3-4-1 es equivalente al 1-4-3-2-1, por consiguiente la cantidad neta de posibilidades para el caso simétrico es $\frac{(n-1)!}{2}$.

De lo anterior, y como es característico en los problemas combinatorios, se tiene que para el TSP el espacio de soluciones factibles crece desbordantemente frente a pequeños aumentos del tamaño del problema, como se puede ver en la Tabla 1 en la que en las primeras tres filas el aumento de n es de cinco en cinco, mientras que la cantidad de soluciones crece a un ritmo astronómico, y en la última fila se ve que un pequeño aumento de 1, es decir, pasar de 15 a 16, lleva al espacio factible a dimensiones del orden de $6,53837E+11$, lo cual es gigantesco. A este fenómeno de crecimiento tan desproporcionado se le denomina *explosión combinatoria* y es lo que complejiza el problema, ya que éste tiene un planteamiento sencillo, pero lo que lo sugiere un reto es la búsqueda de la solución óptima en espacios factibles tan grandes (Bjorndal et al., 1995; Laporte, 2010).

Tabla 1. Explosión combinatoria del TSP

Cantidad de nodos [n]	Cantidad de soluciones factibles
5	12
10	181.440
15	43.589.145.600
16	6,53837E+11

Elaboración propia.

En el marco de esta dificultad para la resolución del problema, muchos investigadores se han interesado en el desarrollo de estrategias de solución (Sun, 2020). Una de las primeras propuestas fue la de Dantzig, Fulkerson, & Johnson en 1954, planteando el uso de la programación matemática en la abstracción del problema, y aprovechando así los desarrollos sobre la solución de este tipo de modelos para llegar a optimizar la distancia total recorrida planteada en la función objetivo de dicho modelo matemático. Basándose en ese planteamiento, Miller, Tucker & Zemlin en 1960 proponen una simplificación de las restricciones para el rompimiento de sub-tours del modelo de Dantzig et al., y a este nuevo planteamiento se le conoce en la literatura como el modelo MTZ en reconocimiento a sus autores, el cual se clasifica como un modelo de programación lineal entera mixta, como se puede evidenciar a continuación:

$$\min Z = \sum_{i \in V} \sum_{j \in V, i \neq j} d_{ij} * x_{ij} \quad (1)$$

$$\sum_{i \in V, i \neq j} x_{ij} = 1, \quad \forall j \in V \quad (2)$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1, \quad \forall i \in V \quad (3)$$

$$u_i - u_j + nx_{ij} \leq n - 1, \quad \forall i, j \in V; i \neq j; i > 0; j > 0 \quad (4)$$

Donde x_{ij} es una variable binaria (0-1) que con un 1 indica que dentro del tour hay un viaje desde i hasta j , y con un 0 indica que no. u_i es una variable de nivel para el rompimiento de sub-tours y corresponde a un número arbitrario mayor o igual que cero. d_{ij} es la matriz de distancias entre el nodo i y el nodo j , previamente conocida. Así pues, en (1), Z corresponde a la función de la distancia total recorrida que se pretende minimizar, (2) es la restricción que indica que todos los nodos deben visitarse, es decir, que deben tener arcos de entrada y además esa vista debe ser única. Similarmente (3) garantiza que de cada nodo se debe salir una única vez, y por último, (4) es la restricción para el rompimiento de sub-tours.

Se puede observar cómo la simplicidad del enunciado del problema mencionada por Laporte (2010) se ve reflejada en la estructura matemática del modelo MTZ ((1)-(4)), que también es relativamente simple. No obstante, y a pesar de la actual potencia de solvers que optimizan modelos enteros mixtos de tamaños cada vez mayores, la dimensión de los problemas de ruteo que se enfrentan en contextos reales, desborda la capacidad actual de estos solvers, impidiendo la consecución de la solución óptima en tiempos razonables (Bjorndal et al., 1995).

Esa necesidad práctica de poder solucionar problemas de gran escala, unida al gran interés científico motivado por el contraste entre la simplicidad en el planteamiento y la complejidad computacional del problema (Laporte, 2010), ha puesto al TSP como uno de los pocos problemas de la investigación de operaciones que ha tenido tanta colaboración entre el ámbito práctico y el académico (Bjorndal et al., 1995), lo que ha impulsado la investigación y el estudio de este problema combinatorio, especialmente frente al desarrollo de nuevos algoritmos que mejoren la eficiencia en la solución.

3.3 TIME DEPENDENT TRAVELING SALESMAN PROBLEM (TDTSP)

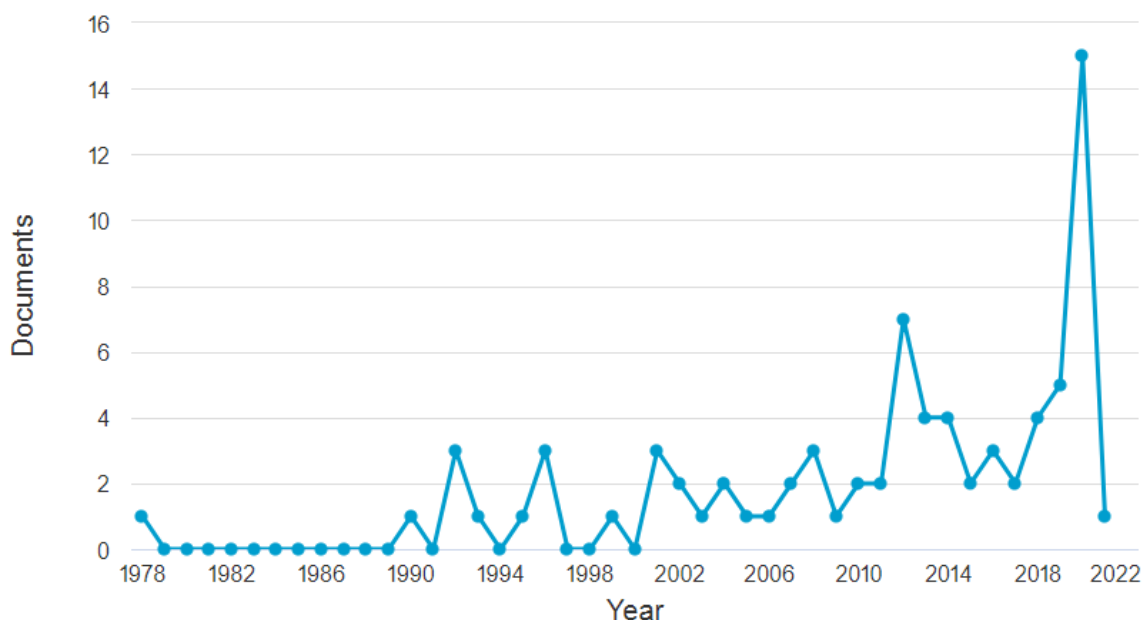
En esta sección se realiza una caracterización del problema de ruteo tiempo dependiente considerando elementos desde la conceptualización del problema presentados en trabajos básicos como los de (Fox, Gavish, & Graves, 1980; Picard & Queyranne, 1978), hasta planteamientos matemáticos posteriores como los de (Hashimoto et al., 2008; Malandraki & Daskin, 1992; Orejuela & Hernández, 2019). Y a partir de dicho marco teórico se establece una representación matemática del problema de esta tesis, dando así cumplimiento al primer objetivo específico.

En la literatura científica se identifican variaciones del TSP, entre las cuales se encuentra el caso en que el tiempo de viaje entre cada par de nodos (i, j) no es un parámetro fijo, sino que es función de la hora del día en que ese recorrido se realice (Hashimoto et al., 2008). Esta generalización del TSP en donde se considera una dependencia respecto al tiempo, es un fenómeno muy común en los ambientes reales de viajes urbanos en donde la congestión en horas picos es evidente, y se denomina como Time Dependent Traveling Salesman Problem (TDTSP) (Malandraki & Daskin, 1992).

El problema de ruteo tiempo dependiente o time-dependent traveling salesman problem (TDTSP), es una generalización del TSP, que debido a su aplicación en una amplia gama de problemas reales ha sido objeto de investigación científica (Gouveia & Voß, 1995). El problema fue introducido por Fox (1973) en un contexto manufacturero de programación de la producción (scheduling) y su enunciación básica como un problema de scheduling consiste en minimizar el costo del procesamiento de n trabajos en una única máquina (single machine), con la particularidad de que los costos de alistamiento de cada trabajo no dependen únicamente del predecesor, sino también de la posición (tiempo) en la secuencia.

Para tener una referencia de la evolución del problema en la literatura científica, se realiza una búsqueda en la base de datos referencial Scopus el día 19 de enero del 2021, con la siguiente ecuación de búsqueda: *TITLE-ABS-KEY ("time dependent traveling salesman problem")*, obteniendo como resultado 78 documentos distribuidos anualmente como muestra la Figura 9 donde también se observa el creciente interés en este tema de investigación en los últimos años, especialmente en el 2020.

Figura 9. Documentos por año del TDTSP en Scopus

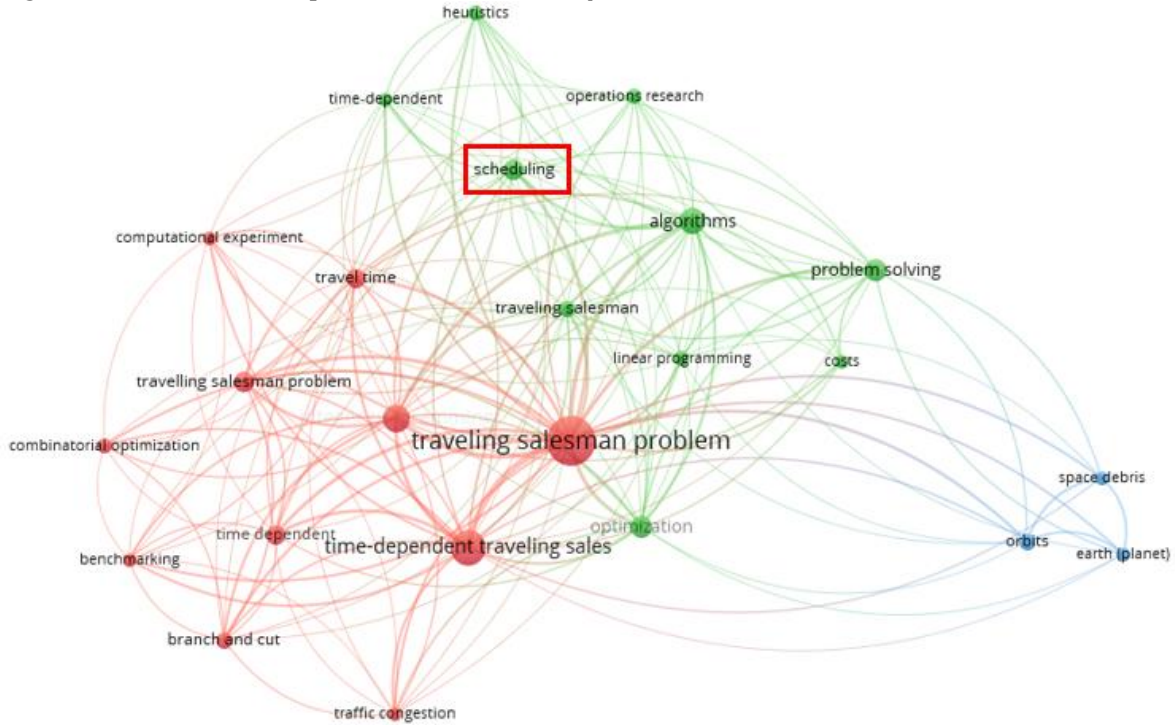


Fuente: Scopus

Con esta búsqueda se realiza también un análisis de co-ocurrencia de palabras clave con el software VOSviewer, obteniendo el gráfico de la Figura 10 en la que se pueden observar tres clúster marcados, el clúster rojo que fundamentalmente hace referencia a las características del problema en donde se encuentran palabras clave como tiempo dependiente, tiempos de viaje y congestión vial. El clúster verde que clasifica las palabras clave relacionadas con las estrategias de solución, en donde se resaltan palabras como optimización, heurísticas, algoritmos y programación lineal; al respecto resulta interesante observar que el nodo de mayor relevancia dentro de dicho clúster verde es el de algoritmos, sugiriendo la complejidad del problema que requiere sus usos. Por último el clúster azul es un clúster aislado referido a ciertas aplicaciones en el ámbito aeroespacial.

Cabe anotar que se realizó la misma búsqueda en Web of Science, con la ecuación: *ALL="time dependent traveling salesman problem"*, y solamente se encontraron 22 documentos relacionados con el TDTSP, lo cual sugiere que es un problema con potencial de investigación.

Figura 10. Co-ocurrencia palabras claves en búsqueda del TDTSP



Fuente: software VOSviewer

En la Figura 10 también se puede observar la relación existente entre los problemas combinatorios de scheduling y ruteo, que conceptualmente radica en que ambos problemas buscan secuenciar, ya sean trabajos o nodos a visitar en una ruta. Y precisamente esa relación del scheduling con el TDTSP se trata desde los trabajos básicos como el artículo de Picard & Queyranne (1978) en el que se plantean dos formulaciones para el TDTSP, entendido como una generalización del TSP en el que el costo de cada trayecto depende no solamente de las respectivas localizaciones involucradas, sino también de la posición en la secuencia que define el tour.

El planteamiento del problema en ese artículo se da en términos del contexto de los problemas de scheduling, en el que se considera un conjunto de trabajos los cuales deben ser procesados por una máquina (one-machine). Los costos de alistamiento C_{ij}^t ocurren cuando el trabajo i es procesado en la t -ésima posición en la secuencia, y cuyo sucesor es el trabajo j (procesado en la posición $t+1$). La máquina tiene un estado inicial antes de que el procesamiento de los trabajos inicie, el cual se asume como dado y se denota como 0. Similarmente, la máquina debe retornar a un estado final después de que el procesamiento de los trabajos finalice, que se asume dado y que se denota como $n+1$. También se asumen como dados los costos de alistamiento inicial y final C_{0i}^0 y C_{in+1}^n . El problema es, entonces, encontrar la secuencia $w(1), w(2), \dots, w(n)$ que minimice el costo total de alistamiento C_w definido como:

$$C_w = C_{0w(1)}^0 + C_{w(1)w(2)}^1 + \dots + C_{w(n-1)w(n)}^{n-1} + C_{w(n)n+1}^n$$

Para este problema los autores proponen principalmente dos formulaciones: una generalización a las formulaciones clásicas del TSP en donde ahora la matriz de distancias es tiempo dependiente, y para lo cual se plantea el uso de tres índices; y una formulación basada en un modelo de asignación

cuadrática, en donde se interpreta el TDTSP como una instancia especial del ya conocido problema de asignación cuadrática (QAP, por sus siglas en inglés Quadratic Assignment Problem).

Formulación de tres índices de Picard & Queyranne (3PQ):

Conjuntos:

Estados máquina = $\{0, 1, \dots, n, n+1\}$, *índices* $\rightarrow i, j, k$

Posición secuencia = $\{0, 1, \dots, n\}$, *índices* $\rightarrow t$

Parámetros:

$c_{ij}^t \equiv$ Costo de pasar del estado i en la posición t al estado j (en la posición $t+1$).

Variables:

$X_{ij}^t \equiv \begin{cases} 1, & \text{si se pasa del estado } i \text{ en la posición } t \text{ al estado } j \text{ (en la posición } t+1) \\ 0, & \text{sino} \end{cases}$

Función objetivo:

$$\text{Min } Z = \sum_{j=1}^n c_{0j}^0 X_{0j}^0 + \sum_{t=1}^{n-1} \sum_{i=1}^n \sum_{j=1, i \neq j}^n c_{ij}^t X_{ij}^t + \sum_{i=1}^n c_{in+1}^n X_{in+1}^n \quad (5)$$

Restricciones:

$$\sum_{j=1}^n X_{0j}^0 = 1 \quad (6)$$

$$X_{0j}^0 = \sum_{k=1, k \neq j}^n X_{jk}^1; \quad \forall j = 1, \dots, n \quad (7)$$

$$\sum_{i=1, i \neq j}^n X_{ij}^t = \sum_{k=1, k \neq j}^n X_{jk}^{t+1}; \quad \forall t = 1, \dots, n-2; \quad \forall j = 1, \dots, n \quad (8)$$

$$\sum_{i=1, i \neq j}^n X_{ij}^{n-1} = X_{jn+1}^n; \quad \forall j = 1, \dots, n \quad (9)$$

En este modelo se observa que la matriz de distancias o costos es una matriz tridimensional con un índice de tiempo, lo cual induce que la decisión involucre también el aspecto temporal y conlleva a que las restricciones representen la dinámica de evolución temporal como se observa en (7)-(9), lo cual es una característica fundamental para los modelos del TDTSP.

Formulación basada en el QAP de Picard & Queyranne (1978):

Básicamente el problema de asignación cuadrática (QAP) consiste en asignar n instalaciones en n localizaciones de tal manera que se minimice el costo de asignación (Neos Guide, n.d.), para lo cual la formulación clásica define la variable X_{ij} como una matriz cuadrada ($n \times n$) cuyas filas corresponden a las instalaciones y las columnas a las localizaciones, de manera tal que $X_{ij} = 1$ significa que la instalación i se asigna a la localización j , de lo contrario se tendría $X_{ij} = 0$.

En este orden de ideas, para el QAP debe cumplirse que cada instalación se asigne a una única localización, y que cada localización solamente tenga asignada una única instalación, lo cual se expresa matemáticamente como sigue:

$$\sum_{i=1}^n X_{ij} = 1; \quad \forall j = 1, 2, \dots, n$$

$$\sum_{j=1}^n X_{ij} = 1; \quad \forall i = 1, 2, \dots, n$$

Volviendo al TDTSP, Picard & Queyranne (1978) plantean una interpretación al problema desde el paradigma del QAP, en donde las instalaciones corresponden a los trabajos y las localizaciones a las posiciones en la secuencia, de forma que el problema consiste en “asignar” los trabajos a las posiciones en la secuencia. Para esto se requiere, entonces, definir el análogo del costo de asignación que para el caso del TDTSP es el costo tiempo dependiente de los trayectos, el cual se expresa desde el paradigma del QAP como un parámetro de cuatro índices a_{ijpq} que es el costo de pasar del trabajo i en la posición p al trabajo j en la posición q , parámetro análogo al de la formulación de tres índices (3PQ) c_{ij}^t en donde el momento de procesamiento del trabajo j ($t+1$) está implícito, es decir, $a_{ijtt+1} = c_{ij}^t$. En este orden de ideas, la formulación matemática que proponen los autores es la siguiente:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^n \sum_{q=1}^n a_{ijpq} X_{ip} X_{jq} \quad (10)$$

Sujeto a

$$\sum_{i=1}^n X_{ip} = 1; \quad \forall p = 1, 2, \dots, n \quad (11)$$

$$\sum_{p=1}^n X_{ip} = 1; \quad \forall i = 1, 2, \dots, n \quad (12)$$

$$X_{ij} \in \{0,1\}; \quad \forall i, p = 1, 2, \dots, n \quad (13)$$

Donde $a_{ijpq} = 0$ excepto por:

$$a_{ii11} = c_{0i}^0 \quad \wedge \quad a_{iinn} = c_{in+1}^n; \quad \forall i$$

$$a_{ijtt+1} = c_{ij}^t; \quad \forall i, j, t \quad (i \neq j \wedge 1 \leq t \leq n-1)$$

Gouveia & Voß (1995) consideran el artículo de Picard & Queyranne (1978), descrito anteriormente, como uno de los trabajos básicos sobre el TDTSP, y hacen una clasificación de las formulaciones que hasta ese momento se planteaban en la literatura respecto a dicho problema, en donde se destacan principalmente una linealización del modelo de asignación cuadrática de Picard & Queyranne (1978), y una formulación de 3 índices planteada por Fox, Gavish, & Graves (1980), que se presentan a continuación.

Este artículo de Gouveia & Voß (1995) parte de la interpretación del problema como un conjunto de n nodos que deben ser visitados una única vez y cuyo costo de viaje entre el nodo i y el nodo j depende de la posición del nodo i en la secuencia. Así, considerando una solución factible al problema como un grafo, se tiene que si el nodo i es el t -ésimo nodo en la secuencia, y el nodo j es el $(t+1)$ -ésimo nodo en el tour, entonces dichos nodos están conectados por el arco (i, j) que es el t -ésimo arco del

tour. Por consiguiente, además del conjunto de nodos, toma relevancia también el conjunto de arcos cuya cardinalidad es igual a la cantidad de nodos del problema.

El problema es encontrar la secuencia de nodos cuyo costo total de viaje sea el mínimo, asumiendo, sin pérdida de generalidad, que el tour parte del nodo 1 con el arco 1 y retorna al nodo mismo nodo 1 con el arco n . A continuación, como marco de referencia, se presentan dos formulaciones discutidas en el artículo de Gouveia & Voß (1995).

Linealización del modelo de asignación cuadrática de Picard & Queyranne (1978):

Este modelo se basa en (10)-(13) involucrando un conjunto de variables Y_{ijt} con el objetivo de linealizar la formulación.

Conjuntos:

Nodos = $\{1, 2, \dots, n\} \rightarrow i, j$

Arcos = $\{1, 2, \dots, n\} \rightarrow t$

Parámetros:

$c_{ijt} \equiv$ costo de viaje del nodo i al nodo j en el arco t .

Variables:

$X_{it} \equiv \begin{cases} 1, & \text{si se parte del nodo } i \text{ en el arco } t \\ 0, & \text{sino} \end{cases}$

$Y_{ijt} \equiv \begin{cases} 1, & \text{si se va del nodo } i \text{ al nodo } j \text{ en el arco } t \\ 0, & \text{sino} \end{cases}$

Función objetivo:

$$\text{Min } Z = \sum_{i=2}^n c_{1i1} X_{i2} + \sum_{i=2}^n \sum_{j=2}^n \sum_{t=2}^{n-1} c_{ijt} Y_{ijt} + \sum_{i=2}^n c_{i1n} X_{in} \quad (14)$$

Restricciones:

$$\sum_{t=2}^n X_{it} = 1; \quad \forall i = 2, \dots, n \quad (15)$$

$$\sum_{i=2}^n X_{it} = 1; \quad \forall t = 2, \dots, n \quad (16)$$

$$\sum_{i=2}^n \sum_{j=2}^n \sum_{t=2}^{n-1} Y_{ijt} = n - 2 \quad (17)$$

$$X_{it} + X_{j(t+1)} - 2Y_{ijt} \geq 0; \quad \forall i, j = 2, \dots, n; \quad \forall t = 2, \dots, n - 1 \quad (18)$$

$$X_{it} \in \{0, 1\}; \quad \forall i, t = 2, \dots, n \quad (19)$$

$$Y_{ijt} \in \{0, 1\}; \quad \forall i, j = 2, \dots, n; \quad \forall t = 2, \dots, n - 1 \quad (20)$$

En la formulación (14)-(20) se simplifica la indexación de la variable Y_{ijt} en la que $i, j = 2, \dots, n$ y $t = 2, \dots, n - 1$, dado que la partida del nodo 1 (depósito) y la llegada a este, se modelan con la variable X_{it} . Es decir, la variable Y_{ijt} representa todos los arcos del tour, excepto el primero y el último, tal como se expresa en (17). En (18) se relacionan las variables X_{it} y Y_{ijt} , de manera que si el arco (i, j) es el t -ésimo del tour ($Y_{ijt} = 1$), entonces el nodo i es el t -ésimo de la secuencia ($X_{it} =$

1) y el nodo j es el $(t+1)$ -ésimo ($X_{j(t+1)} = 1$). Y esta restricción (18) es la que modela la secuencia temporal del problema, característica fundamental de los problemas tiempo dependiente.

Formulación de 3 índices de Fox, Gavish, & Graves (1980):

Esta formulación de 3 índices es posterior a la 3PQ ((5)-(9)) y los autores la presentan con la ventaja de requerir únicamente $4n$ restricciones lineales.

Conjuntos:

Nodos = $\{1, 2, \dots, n\} \rightarrow i, j$

Arcos = $\{1, 2, \dots, n\} \rightarrow t$

Parámetros:

$c_{ijt} \equiv$ costo de viaje del nodo i al nodo j en el arco t .

Variables:

$Y_{ijt} \equiv \begin{cases} 1, & \text{si se va del nodo } i \text{ al nodo } j \text{ en el arco } t \\ 0, & \text{sino} \end{cases}$

Función objetivo:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^n c_{ijt} Y_{ijt} \quad (21)$$

Restricciones:

$$\sum_{i=1}^n \sum_{t=1}^n Y_{ijt} = 1; \quad \forall j = 1, 2, \dots, n \quad (22)$$

$$\sum_{j=1}^n \sum_{t=1}^n Y_{ijt} = 1; \quad \forall i = 1, 2, \dots, n \quad (23)$$

$$\sum_{i=1}^n \sum_{j=1}^n Y_{ijt} = 1; \quad \forall t = 1, 2, \dots, n \quad (24)$$

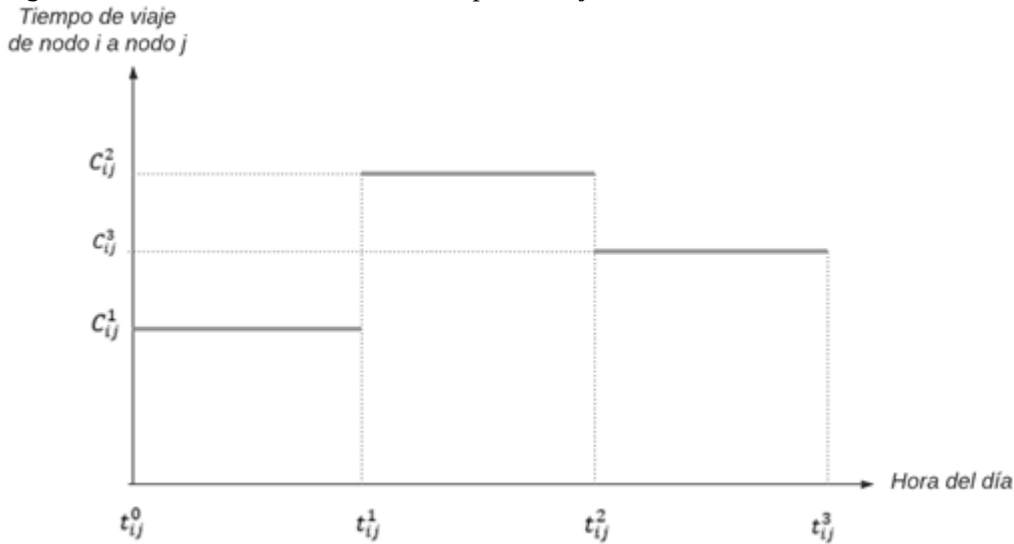
$$\sum_{j=1}^n \sum_{t=2}^n t Y_{ijt} - \sum_{j=1}^n \sum_{t=1}^{n-1} t Y_{jit} = 1; \quad \forall i = 2, 3, \dots, n \quad (25)$$

La formulación se presenta originalmente en (Fox et al., 1980) y su aporte radica en la restricción (25) que representa la dinámica temporal del problema y que como se argumenta en el Lema 1 del artículo, también garantiza el rompimiento de sub-tours. La lógica de la restricción parte del hecho de que para un i arbitrario entre 2 y n , si se llega a él a través del arco t , entonces el arco del que se sale de dicho nodo i debe ser el arco $t+1$.

Tanto las formulaciones de Picard & Queyranne (1978), como la linealización del modelo de asignación cuadrática presentado en Gouveia & Voß (1995), así como también la formulación de tres índices de Fox, Gavish, & Graves (1980), parten de la consideración del tiempo desde una perspectiva discreta, en donde el costo de viaje entre dos nodos depende de la posición en la secuencia, suponiendo que el avance del tiempo corresponde al avance discreto en la secuencia. Sin embargo, en formulaciones posteriores, se relaja ese supuesto de que el costo de viaje depende de la posición en la secuencia, más se considera una perspectiva continua del tiempo en el que el costo de viaje depende de la hora del día.

En este sentido se encuentra el artículo de Malandraki & Daskin (1992) en el que el TDTSP es formulado como un programa lineal entero mixto para el caso en que el tiempo de viaje entre dos nodos (i, j) es una función del tiempo en el que se parte del nodo i (denotado como t_i). Dicha función $c_{ij}(t_i)$ se define como una función escalonada en la que el día es dividido en intervalos de tiempo, de manera que si se conoce el instante de tiempo en que se parte del nodo i al nodo j , entonces también se conoce el tiempo de viaje de ir de i a j , el cual es una constante para el intervalo de tiempo, como se observa en la Figura 11, en donde se tienen tres intervalos de tiempo definidos a través del parámetro t_{ij}^k , donde el índice k opera sobre el conjunto de intervalos, de manera que dicho parámetro es el límite superior del intervalo k para el arco (i, j) . Nótese que en esta lógica, el límite inferior de un intervalo arbitrario k , se define como el límite superior del intervalo $k-1$. Cabe notar también que la función del tiempo de viaje en (i, j) , es un parámetro también indexado en k (c_{ij}^k) y cuyo valor es constante para cada k .

Figura 11. Función escalonada del tiempo de viaje



Fuente: (Malandraki & Daskin, 1992)

El modelo de Malandraki & Daskin (1992) considera ventanas de tiempo en los nodos, pero dado que en la presente tesis no se van a considerar ventanas de tiempo, a continuación se presenta una simplificación del modelo en donde las estructuras asociadas a las ventanas de tiempo se omiten, dejando únicamente lo concerniente a las dinámicas tiempo dependientes de los tiempos de viaje entre nodos. En esta formulación el depósito se interpreta como el nodo 1 cuando inicia el tour, y el nodo $n+1$ cuando finaliza el tour.

Conjuntos:

Nodos $[N] = \{1, 2, \dots, n, n+1\} \rightarrow i, j$

Intervalos de tiempo $[M] = \{1, 2, \dots, m\} \rightarrow k$

Parámetros:

$c_{ij}^k \equiv$ tiempo de viaje del nodo i al nodo j si la partida de i ocurre en intervalo k .

$c_i \equiv$ tiempo de servicio en el nodo i .

$t_{ij}^k \equiv$ límite superior del intervalo k para el arco (i, j) .

$t \equiv$ tiempo de inicio en el tour.

$B1 \equiv$ número muy grande.

$B2 \equiv$ número muy grande.

Variables:

$$X_{ij}^k \equiv \begin{cases} 1, & \text{si se viaja del nodo } i \text{ al nodo } j \text{ partiendo de } i \text{ en el intervalo } k \\ 0, & \text{sino} \end{cases}$$

$t_j \equiv$ tiempo de partida del nodo j .

Función objetivo:

$$\text{Min } Z = t_{n+1} \quad (26)$$

Restricciones:

$$\sum_{i=1, i \neq j}^n \sum_{k=1}^m X_{ij}^k = 1; \quad \forall j = 2, 3, \dots, n+1 \quad (27)$$

$$\sum_{j=2, j \neq i}^{n+1} \sum_{k=1}^m X_{ij}^k = 1; \quad \forall i = 1, 2, \dots, n \quad (28)$$

$$t_1 = t \quad (29)$$

$$t_j - t_i - B1 * X_{ij}^k \geq c_{ij}^k + c_j - B1; \quad i = 1, \dots, n; j = 2, \dots, n+1; k = 1, \dots, m; i \neq j \quad (30)$$

$$t_i + B2 * X_{ij}^k \leq t_{ij}^k + B2; \quad i = 1, \dots, n; j = 2, \dots, n+1; k = 1, \dots, m; i \neq j \quad (31)$$

$$t_i - t_{ij}^{k-1} * X_{ij}^k \geq 0; \quad i = 1, \dots, n; j = 2, \dots, n+1; k = 1, \dots, m; i \neq j \quad (32)$$

$$c_i = 0; \quad i = 1, n+1 \quad (33)$$

$$c_{ii}^k = \infty; \quad i = 1, \dots, n+1; k = 1, \dots, m \quad (34)$$

En este modelo el conjunto de restricciones (30) representan la dinámica de evolución del tiempo a través de las variables t_j , donde si se va del nodo i al nodo j , entonces la diferencia entre el tiempo de partida del nodo j y el tiempo de partida del nodo i es el tiempo de viaje más el tiempo de servicio en el nodo j ; éste control del tiempo es la característica fundamental de los problemas tiempo dependientes. Por otro lado, al considerar el tiempo desde una perspectiva continua, y al definir intervalos de tiempo, es necesario modelar la asignación de la variable de tiempo t_i con su respectivo intervalo de tiempo k , lo cual se realiza a través de las restricciones (31) y (32) usando los parámetros de los límites de los intervalos (t_{ij}^k).

Una particularidad del modelo (26)-(34) es el manejo del tiempo de inicio del tour, que en este caso se considera como dado a través del parámetro t , es decir, el modelo no decide el tiempo de inicio del tour, sino que parte de dicho tiempo como conocido (restricción (29)) y decide solamente sobre la secuencia de visitas. Sin embargo, para algunos autores como (Hashimoto et al., 2008; Orejuela & Hernández, 2019) al tratarse de un problema tiempo dependiente, el elemento temporal debe considerarse también como una decisión. Y en este sentido, Osorio, Bravo-Bastidas, & Orejuela (2021) muestran la importancia del tiempo de inicio del tour en el comportamiento de un modelo de ruteo con dependencias temporales y la sensibilidad de los resultados frente a cambios en dicho tiempo, mostrando así la relevancia de considerarlo como un elemento decisorio en la modelación.

Considerando esa importancia de la determinación del tiempo de inicio del tour en el TDTSP, Hashimoto, Yagiura, & Ibaraki (2008) proponen la división del problema en: primeramente, determinar la secuencia de nodos a visitar y posteriormente, dada la ruta, determinar el tiempo de

inicio óptimo. Ese sub-problema del tiempo de inicio, los autores lo representan con un modelo matemático en el marco de un problema genérico de ruteo con varios vehículos capacitados, y considerando ventanas de tiempos en los nodos, sin embargo, como marco de referencia para esta tesis, se va a considerar una simplificación del modelo en donde solamente se considerarán los tiempos de viaje tiempo dependientes y sus dinámicas.

En este modelo se hace una abstracción del tiempo desde una perspectiva continua, se considera el depósito como el nodo 0, y se asume la secuencia de vistas como parámetro, denotada como σ , que es un vector con $n+2$ componentes, siendo n la cantidad de nodos a visitar, donde la primera componente es el nodo 0, así como también la última componente, ya que el tour debe iniciar y terminar en el depósito. A continuación se escribe formal y simbólicamente el modelo de Hashimoto, Yagiura, & Ibaraki (2008).

Conjuntos:

$Nodos[N] = \{0, 1, \dots, n\} \rightarrow i, j$

Parámetros:

$\lambda_{ij}(T) \equiv$ Función de tiempo de viaje de i a j , dependiente de tiempo de partida de i .

$\sigma \equiv$ Ruta del vehículo; $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n+2))$, donde $\sigma(1) = \sigma(n+2) = 0$

Variables:

$T_i \equiv$ tiempo de partida del nodo i

$S_i \equiv$ tiempo de inicio del servicio en nodo i

Función objetivo:

$$\text{Min } Z = \sum_{h=1}^{n+1} \lambda_{\sigma(h), \sigma(h+1)}(T_{\sigma(h)}) \quad (35)$$

Restricciones:

$$S_i \leq T_i; \quad \forall i \in N \setminus \{0\} \quad (36)$$

$$T_{\sigma(h)} + \lambda_{\sigma(h), \sigma(h+1)}(T_{\sigma(h)}) \leq S_{\sigma(h+1)}; \quad 1 \leq h \leq n+1 \quad (37)$$

Es importante notar que en este modelo ((35)-(37)) el tiempo de inicio del tour (T_0) es una variable de decisión y que el modelo se plantea precisamente para representar el problema de encontrar el valor de T_0 óptimo. Las restricciones modelan particularmente la evolución temporal del problema, en donde se busca garantizar que si se parte del nodo i al nodo j , entonces el tiempo de llegada al nodo j debe ser mayor o igual al tiempo de partida del nodo i más el tiempo de viaje en función de ese tiempo de partida de i (restricción (37)). Como se evidencia en los modelos presentados anteriormente, ese control temporal es característico en los problemas tiempo dependientes.

También es importante tener en cuenta que los autores plantean la división del problema como una estrategia de solución en el marco de un algoritmo que usa búsqueda local para determinar las rutas, y para la optimización del tiempo de inicio se utiliza programación dinámica dentro del mismo algoritmo propuesto. Dicha separación es una idea muy interesante en el marco de las estrategias de solución al problema y posteriormente se va a profundizar más al respecto, sin embargo, en este punto se pretende establecer la formulación matemática del problema en esta tesis, a partir del marco teórico presentado anteriormente y como cumplimiento al primer objetivo específico.

En este orden de ideas, la formulación planteada en esta tesis va a considerar el tiempo desde la perspectiva continua, definiendo intervalos de tiempo, siguiendo a Malandraki & Daskin (1992) (modelo (26)-(34)). Respecto al tiempo de inicio del tour, se sigue la idea de Hashimoto, Yagiura, & Ibaraki (2008) incorporándolo como una variable de decisión en el modelo. Y finalmente, las estructuras que rigen las dinámicas del tiempo se basan en el modelo propuesto por Orejuela & Hernández (2019). El modelo que se propone es entonces el siguiente:

Conjuntos:

$Nodos[N] = \{0, 1, \dots, n\} \rightarrow i, j$

$Periodos[P] = \{1, 2, \dots, m\} \rightarrow k$

Parámetros:

$t_{ijk} \equiv$ tiempo de viaje del nodo i al nodo j en el periodo k

$ts_i \equiv$ tiempo de servicio en el nodo i

$ip_k \equiv$ inicio del periodo k

$fp_k \equiv$ fin del periodo k

$M \equiv$ número muy grande

Variables:

$X_{ijk} \equiv \begin{cases} 1, & \text{si se viaja del nodo } i \text{ al nodo } j \text{ en el periodo } k \\ 0, & \text{sino} \end{cases}$

$T_i \equiv$ tiempo de partida del nodo i

Función objetivo:

$$\text{Min } Z = \sum_{i \in N} \sum_{j \in N, i \neq j} \sum_{k \in P} X_{ijk} t_{ijk} \quad (38)$$

Restricciones:

$$\sum_{i \in N, i \neq j} \sum_{k \in P} X_{ijk} = 1; \quad \forall j \in N \quad (39)$$

$$\sum_{j \in N, i \neq j} \sum_{k \in P} X_{ijk} = 1; \quad \forall i \in N \quad (40)$$

$$T_i + \sum_{k \in P} X_{ijk} t_{ijk} \leq T_j - ts_j + \left[1 - \sum_{k \in P} X_{ijk} \right] * M; \quad \forall i, j \in N, i \neq j, j \neq 0 \quad (41)$$

$$T_i + \sum_{k \in P} X_{ijk} t_{ijk} + \left[1 - \sum_{k \in P} X_{ijk} \right] * M \geq T_j - ts_j; \quad \forall i, j \in N, i \neq j, j \neq 0 \quad (42)$$

$$\sum_{j \in N, i \neq j} \sum_{k \in P} X_{ijk} ip_k \leq T_i; \quad \forall i \in N \quad (43)$$

$$\sum_{j \in N, i \neq j} \sum_{k \in P} X_{ijk} fp_k \geq T_i; \quad \forall i \in N \quad (44)$$

El modelo busca minimizar el tiempo total del tour tal como se observa en la función objetivo (38). Las restricciones (39) y (40) (análogas a las (2) y (3) del TSP simple) garantizan que a un nodo solo se pueda llegar y salir una única vez. Al ser un modelo para un problema tiempo dependiente, requiere un control estricto de la evolución temporal, lo cual implica agregar complejidad respecto a modelos como el TSP simple planteado en la sección anterior ((1)-(4)). Ahora la decisión no es únicamente respecto a si se realiza el viaje de i a j , sino que es necesario determinar con certeza en qué periodo

de tiempo k se realiza el viaje, es decir, que la variable binaria asociada a la ruta es ahora X_{ijk} . En este orden de ideas, dicha determinación del periodo k en la variable requiere del conocimiento de los tiempos de partida de nodo i , es decir, del control de la evolución del tiempo, lo cual sugiere involucrar en el modelo las variables T_i que corresponden al tiempo de partida del nodo i , además de las estructuras matemáticas que modelen la evolución temporal y sus implicaciones.

El control temporal se rige por las restricciones (41) y (42) garantizando que si se parte del nodo i al nodo j ($\sum_k X_{ijk} = 1$), entonces el tiempo de partida del nodo i más el tiempo de viaje debe ser igual al tiempo de llegada al nodo j ($T_j - ts_j$). Dicho tiempo de viaje es dependiente del periodo k en el que se parte del nodo i , lo cual requiere de restricciones que relacionen la variable X_{ijk} con la variable T_i , considerando los límites de los intervalos de cada periodo k , tal como se observa en (43) y (44).

Comparando el modelo (1)-(4) con el modelo (38)-(44) se evidencia la complejización del planteamiento matemático con la consideración de tiempos dependientes en el TSP, generando más variables y más restricciones con estructuras más robustas. Y si con el modelo simple del TSP básico, como se comentó en el apartado anterior, se tienen limitantes en los tiempos de resolución en instancias grandes, entonces la complejidad causada por las estructuras de control temporal, agudiza aún más la limitante computacional en la resolución del problema. Esta situación se agravaría en mayor magnitud si se piensa que la dependencia temporal no solamente es para los tiempos de recorridos entre nodos, cómo en (38)-(44), sino que los tiempos de servicio también tengan una relación funcional respecto al tiempo (Cacchiani, Contreras-Bolton, & Toth, 2020), lo que conllevaría a un modelo aún más robusto que el que aquí se aborda, empeorando el desempeño de los solvers.

3.4 OPTIMAL START TIME PROBLEM

En el marco del ruteo tiempo dependiente, el sub-problema de determinar el tiempo de inicio óptimo del tour dada la secuencia de visitas, se denomina “problema de tiempo de inicio óptimo”, denotado como OSTP por sus siglas en inglés (Optimal Start Time Problem). Y tal como se menciona en el apartado anterior, este problema fue planteado por Hashimoto, Yagiura, & Ibaraki (2008) como parte de una estrategia de descomposición del problema completo de ruteo para afrontar la complejidad computacional en la resolución.

El modelo que plantean los autores citados resulta ser un programa lineal relativamente sencillo (modelo (35)-(37)). Y siguiendo esta misma lógica de modelamiento, se parte de la formulación planteada en esta tesis para el TDTSP (modelo (38)-(44)), para llegar a un modelo simplificado del OSTP, en donde la variable X_{ijk} ya no es necesaria, toda vez que la secuencia de visitas deja de ser una decisión a tomar, pasando a ser un parámetro del modelo que se denotará como σ .

Conjuntos:

Nodos[N] = $\{0, 1, \dots, n\} \rightarrow i, j$

Periodos[P] = $\{1, 2, \dots, m\} \rightarrow k$

Posición en secuencia[S] = $\{1, 2, \dots, n + 2\} \rightarrow h$

Parámetros:

$t_{ijk} \equiv$ tiempo de viaje del nodo i al nodo j en el periodo k

$ts_i \equiv$ tiempo de servicio en el nodo i

$ip_k \equiv$ inicio del periodo k

$fp_k \equiv$ fin del periodo k

$\sigma \equiv$ secuencia de la ruta, donde $\sigma(1) = \sigma(n+2) = 0$ (depósito)

Variables:

$T_i \equiv$ tiempo de partida del nodo i

$I_{ik} \equiv \begin{cases} 1, & \text{si se parte del nodo } i \text{ en el periodo } k \\ 0, & \text{sino} \end{cases}$

Función objetivo:

$$\text{Min } Z = \sum_{h=1}^{n+1} \sum_{k=1}^m t_{\sigma(h),\sigma(h+1),k} I_{\sigma(h),k} \quad (45)$$

Restricciones:

$$\sum_{k=1}^m I_{\sigma(h),k} = 1; \quad 1 \leq h \leq n+1 \quad (46)$$

$$\sum_{k=1}^m I_{\sigma(h),k} ip_k \leq T_{\sigma(h)}; \quad 1 \leq h \leq n+1 \quad (47)$$

$$\sum_{k=1}^m I_{\sigma(h),k} fp_k \geq T_{\sigma(h)}; \quad 1 \leq h \leq n+1 \quad (48)$$

$$T_{\sigma(h)} + \sum_{k=1}^m t_{\sigma(h),\sigma(h+1),k} I_{\sigma(h),k} = T_{\sigma(h+1)} - ts_{\sigma(h+1)}; \quad 1 \leq h \leq n+1 \quad (49)$$

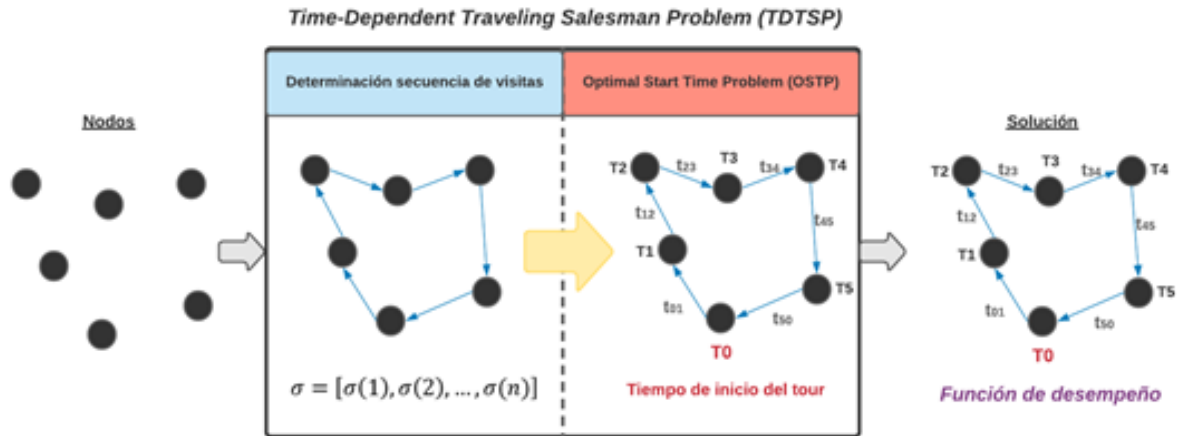
Dado que la dependencia temporal de los tiempos de viaje se relaciona específicamente con el intervalo de tiempo en el que ocurre el viaje (índice k del parámetro t_{ijk}), es necesario definir la variable I_{ik} , que captura el intervalo de tiempo en el que se parte del nodo i . Dicha captura se realiza a través de la variable T_i en las restricciones (47) y (48).

El control de la evolución temporal del problema se realiza con las restricciones (49) y la función objetivo (45) es la misma de la formulación completa del TDTSP, es decir, minimizar el tiempo total del tour.

En términos generales, al observar el modelo para el OSTP ((45)-(49)) es evidente la simplificación respecto a la formulación del TDTSP ((38)-(44)) al reducirse tanto la cantidad de variables como de restricciones, y por tanto, obteniendo un problema con un menor costo computacional de resolución, satisfaciendo así el principio tras la idea de la descomposición, que es llegar a sub-problemas más fáciles de resolver.

En la Figura 12 se muestra una representación gráfica de la descomposición del TDTSP en el sub-problema de la secuenciación de visitas y el del OSTP, en donde se puede observar la relación secuencial entre ambos. Es decir, dado un conjunto de nodos, el TDTSP se puede interpretar en términos de dos fases: la primera fase correspondiente a la exploración de las posibles secuencias de visitas a los nodos, y la segunda fase en la que, para una determinada secuencia, se determinan las dinámicas temporales a partir de la definición del tiempo de inicio óptimo (T_0). Y finalmente, dicha evolución temporal determinará los valores de los parámetros tiempo dependientes que definirán, luego, el valor de la función de desempeño del problema.

Figura 12. El OSTP dentro del TDTSP



Elaboración propia.

Dada la relativa simplicidad del modelo obtenido para el OSTP, éste va a ser de posterior utilidad en el planteamiento metodológico en el que la estrategia a usar contempla el uso de un algoritmo genético, bajo el paradigma de redes de Petri, en donde se encuentre embebido el modelo del OSTP, y así afrontar la exploración de las secuencias de visitas con el genético y luego, involucrar el sub-problema del tiempo de inicio óptimo, dada dichas secuencias.

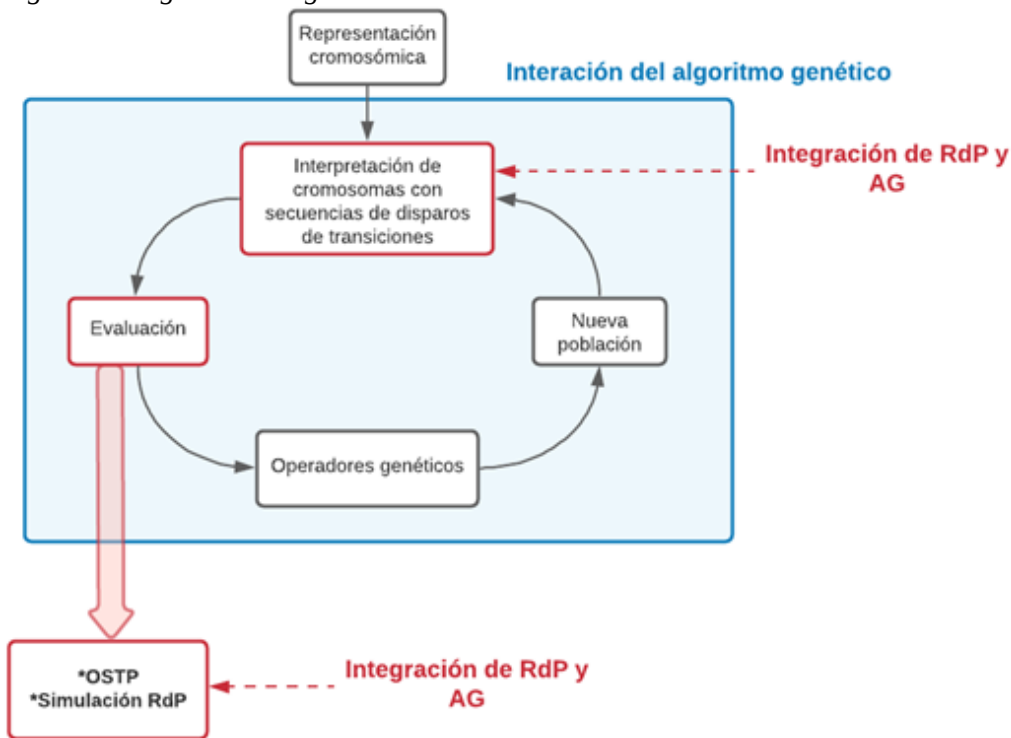
4. PROPUESTA METODOLÓGICA PARA LA SOLUCIÓN DEL TIME-DEPENDENT TRAVELING SALESMAN PROBLEM

Éste es el capítulo en el que se describe el algoritmo propuesto para la solución del TDTSP. Se empieza discutiendo el modelamiento desde el paradigma de redes de Petri diseñando una representación del problema, luego se presenta la lógica de simulación de dicha red propuesta que servirá de base para la integración con el algoritmo genético, permitiendo afrontar el control de la dinámica temporal del problema, y dando cumplimiento con ello al segundo objetivo específico de la tesis.

Posteriormente se aborda el diseño de la propuesta metodológica de solución al TDTSP, basada en un algoritmo que integra el modelo en RdP del problema con un algoritmo genético. En la Figura 13 se ubican los dos puntos de integración de la RdP en el algoritmo genético base. El primer punto corresponde a la etapa de representación de los cromosomas como secuencias de disparos de transiciones, es decir, el problema de secuenciación de visitas se ha traducido al problema de secuenciación de disparos de transiciones. Este paradigma de transiciones permite usar la lógica de simulación de la red en la etapa de evaluación, puesto que dada dicha secuencia de disparos y el tiempo de inicio, la simulación permite determinar el valor de la función de desempeño, y por lo tanto, este es el segundo punto de interacción entre la RdP y el algoritmo genético.

La Figura 13 muestra, también, cómo la propuesta metodológica se fundamenta en la idea de descomposición del TDTSP en las dos fases discutidas en el capítulo anterior (Figura 12). En este sentido, la representación cromosómica y los operadores genéticos permiten explorar la secuencia de visitas (primera fase), mientras que la integración del modelo matemático del OSTP en la etapa de evaluación, permite abordar la segunda fase del problema. A lo largo del capítulo se discute la manera en que se operan el OSTP y la simulación del RdP en la etapa de evaluación.

Figura 13. Algoritmo integrador



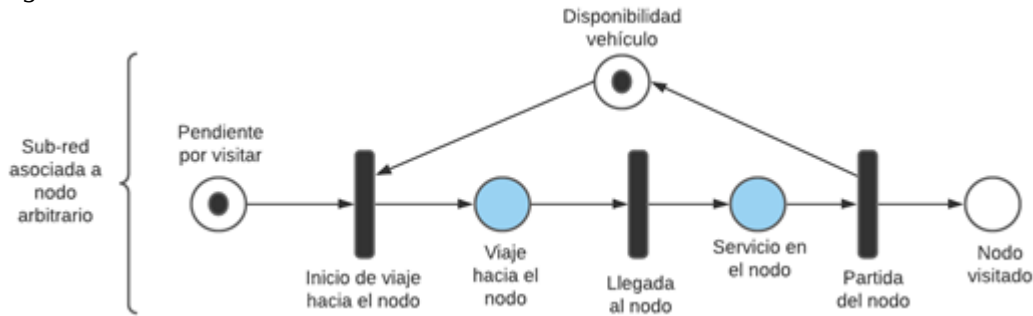
Elaboración propia.

Con el desarrollo del algoritmo propuesto se cumple el tercer objetivo específico de la tesis, y el capítulo finaliza abordando el cuarto objetivo específico, a través de la validación del algoritmo en diversas instancias del problema, proporcionando resultados computacionales que hacen prever esta propuesta como promisoría en aplicaciones futuras.

4.1 Modelamiento propuesto del TDTSP en RdP

El modelamiento de un problema condiciona la posterior forma de solución, así que es pertinente considerar las metodologías que se aplicaran al modelo. Por consiguiente, para la presente tesis el modelo en RdP del problema se compone de n sub-redes, cada una asociada a un nodo a visitar, favoreciendo así la aplicación de algoritmos genéticos. Se trata, además, de una RdP temporizada para poder representar los tiempos de operación relacionados con cada nodo (sub-red), que son el viaje hacia el nodo y el servicio en él, representados con las plazas temporizadas plazas azules de la Figura 14. Por otro lado, las plazas no temporizadas representan los estados de cada nodo: pendiente por visitar o visitado, además de la disponibilidad del vehículo. Las transiciones modelan los eventos de inicio del viaje hacia el nodo, llegada al nodo y partida del nodo, tal como se muestra en la Figura 14.

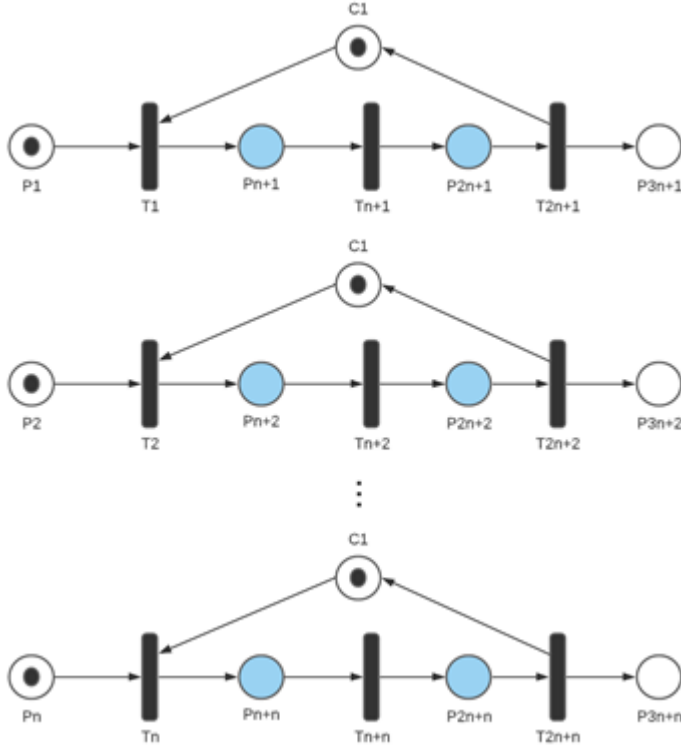
Figura 14. Sub-red



Elaboración propia.

Considerando ahora un problema de n nodos, la estructura completa de la RdP se compone, entonces, de n sub-redes, como se muestra en la Figura 15, cuyos elementos se denotan de manera tal que permitan su identificación con el nodo correspondiente. Así, para el nodo k la sub-red respectiva contiene las siguientes partes: una plaza P_k en la que el token representa que ese nodo k no ha sido visitado, una transición T_k correspondiente al evento de inicio del viaje hacia el nodo k , una plaza P_{n+k} que es temporizada con el tiempo de viaje hacia el nodo k , una transición T_{n+k} que es el evento de llegada al nodo k , una plaza P_{2n+k} cuya temporalización corresponde al tiempo de servicio en el nodo, una transición T_{2n+k} que representa la finalización del servicio en el nodo k y que por tanto, libera la disponibilidad del camión (plaza $C1$), por último una plaza P_{3n+k} en el que el token indica que el nodo k ya ha sido visitado. Cabe anotar que la plaza de disponibilidad del camión ($C1$), es una plaza de recurso que está conectada a cada una de las sub-redes y su objetivo es controlar que ese recurso no pueda ser usado en más de un viaje.

Figura 15. RdP para el TDTSP

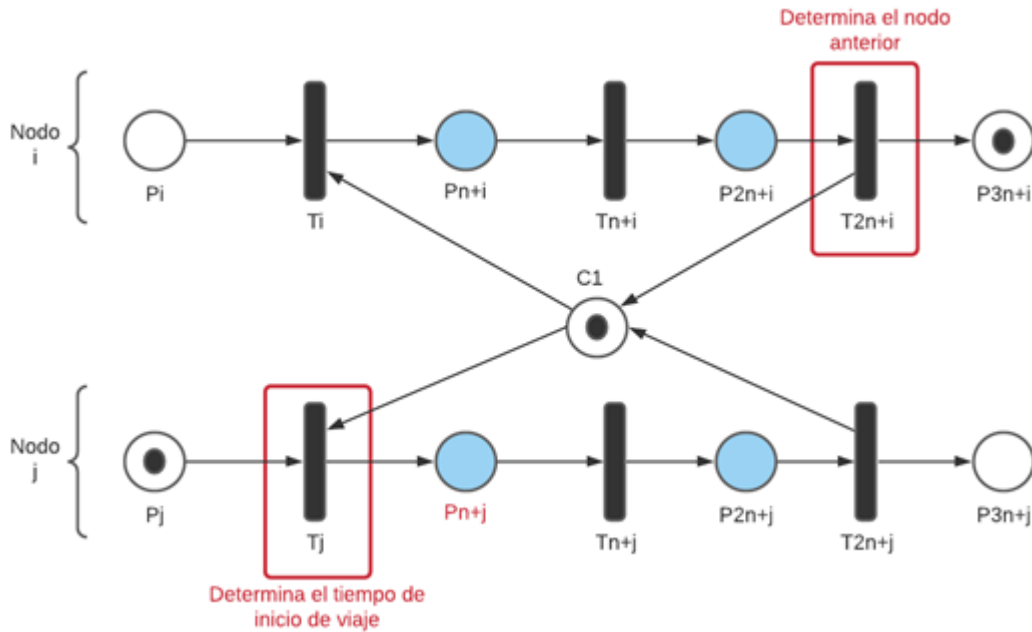


Elaboración propia.

Con este paradigma de modelado, se tiene que para el caso general de n nodos, la RdP se compone de $4n$ plazas asociadas a los nodos, además de la plaza $C1$ asociada a la disponibilidad del camión, es decir, $P = \{1, 2, \dots, 4n + 1\}$. Además de $3n$ transiciones $T = \{1, 2, \dots, 3n\}$. Y con esta estructura, la solución al problema del orden de visitas consiste en la secuenciación de los disparos de las transiciones $T_1, T_2, \dots, T_n$, es decir, los inicios de los viajes a cada nodo, ya que cada T_i ($\forall i \in [1, n]$) tiene una correspondencia unívoca con el nodo i , de manera que secuenciar nodos es equivalente a secuenciar transiciones. Y por otro lado, el control de tiempos se observa con la simulación de la red que va generando la dinámica de disparos de transiciones.

Entonces para dicha simulación de la red se hace necesario saber de antemano la secuencia de transiciones T_i ($i = 1, 2, \dots, n$), además del vector τ que contiene los tiempos asociados a las plazas temporizadas, el cual debe ser un vector dinámico dentro de la simulación, específicamente para los tiempos de viaje, ya que el problema abordado es de naturaleza tiempo dependiente y por consiguiente, el tiempo asociado al viaje hacia un nodo depende tanto del nodo anterior desde el que se parte, como del tiempo de partida. Sin embargo esto no reviste mayor problema en la lógica de la simulación de la red, toda vez que el nodo anterior desde el que se parte se obtiene capturando la última transición T_{2n+k} disparada, donde k es precisamente el nodo anterior que se puede obtener con el índice de la transición, y el tiempo de partida es el valor mismo de la primera transición disparada de la sub-red del nodo. Por ejemplo, suponiendo que se parte del nodo i al nodo j , entonces la parte de la RdP asociada a dicha dinámica es la que se muestra en la Figura 16, donde se marca la transición T_{2n+i} como la última transición disparada antes de que se dispare la transición T_j que corresponde al tiempo de inicio de viaje hacia j , viaje cuyo tiempo es en función de la información que brindan esas dos transiciones, y que, entonces, es el tiempo de viaje asociado a la plaza P_{n+j} .

Figura 16. Dinámica arco (i,j)



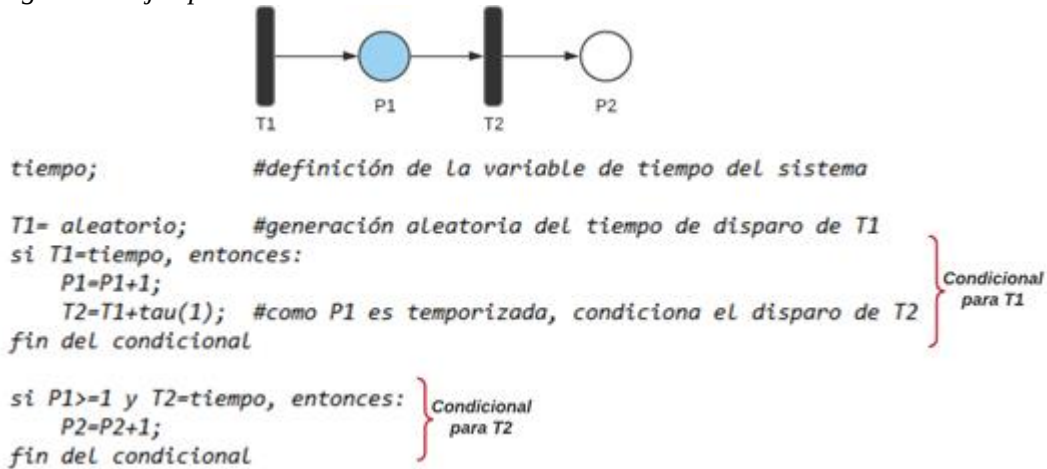
Elaboración propia.

En el siguiente apartado se describe más detalladamente la dinámica de simulación de la red, la cual juega un rol fundamental en el algoritmo integrador de la RdP con un algoritmo genético, que es la propuesta metodológica de esta tesis como solución al problema. En este punto se puede observar la interacción básica entre ambas herramientas: el algoritmo genético explora la secuencia de transiciones T_i ($i = 1, 2, \dots, n$), que siempre será una solución factible del problema, y a partir de dicha secuencia, la simulación de la RdP permite evaluar la función de fitness que es la duración total de tour, y que se captura con la última transición disparada de la red.

Simulación de la RdP para el TDTSP

La simulación de una RdP consiste en la programación de un algoritmo que emule el comportamiento de la red desde la lógica de las reglas de disparo descritas en el apartado 3.1. Básicamente el código consiste en una serie de estructuras condicionales en las que se validan las condiciones necesarias para el disparo de cada transición, y si esto ocurre, se actualiza dentro de la estructura condicional el estado de la red según las reglas de disparo y la evolución del tiempo que marcan las plazas temporizadas. Por ejemplo, en la Figura 17 se muestra una RdP básica con 2 plazas y 2 transiciones, así como el algoritmo en pseudocódigo que simula la red.

Figura 17. Ejemplo simulación RdP



Elaboración propia.

A la luz de la Figura 17 cabe resaltar la dinámica de las plazas temporizadas, que implica relacionar la transición de entrada con la transición de salida, como se observa en el condicional para la transición T1, donde el disparo de dicha transición implica llegar a la plaza temporizada P1 y por tanto define el tiempo de disparo de la transición T2. Esta dinámica va definiendo la evolución del tiempo en la red y es de suma importancia para el caso de la red del TDTSP.

Precisamente respecto al modelo propuesto en esta tesis para el TDTSP es relevante considerar que éste permite simular la dinámica de visitas a los nodos, así como la evolución temporal, siempre y cuando se tenga de antemano tanto la secuencia de vistas a los nodos (secuencia de transiciones T_i , $i=1, 2, \dots, n$), como el tiempo de inicio del tour, ya que bajo estas condiciones iniciales es que la red se dota de la lógica necesaria para simular el sistema real de visitas.

En este orden de ideas, se define la siguiente notación para los elementos del algoritmo de simulación de la RdP: el vector de transiciones se denota como T , el de los tiempos de las plazas como τ , el vector de secuencia de visitas como σ , el tiempo de inicio como T_0 y por último se define una variable UT que va a capturar la última transición disparada en el paso de una sub-red a la otra. Con esta notación definida, se presenta a continuación el algoritmo en pseudocódigo para la simulación de la red, llevando control de los valores de las transiciones, ya que estas brindan la información necesaria de la evolución del tiempo, tan importante para los problemas tiempo dependientes.

La primera parte del algoritmo consiste en la definición de la transición del primer elemento de la secuencia ($\sigma(1)$) con el valor de T_0 , lo cual permite definir el resto de transiciones de la sub-red $\sigma(1)$, como se muestra en la Figura 18, ya que las otras dos transiciones tienen como entrada plazas temporizadas. Y respecto a la plaza $P_{n+\sigma(1)}$ el valor de τ correspondiente está en función del valor de $T_{\sigma(1)}$ (tiempo de inicio del viaje).

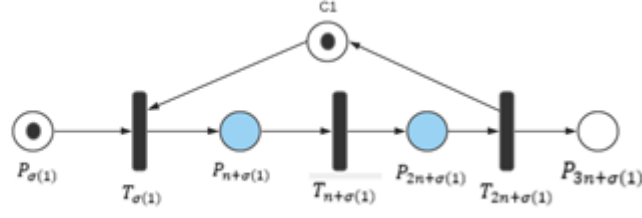
Figura 18. Primera parte algoritmo simulación

$$T_{\sigma(1)} = T_0$$

$$T_{n+\sigma(1)} = T_{\sigma(1)} + \tau_{n+\sigma(1)}$$

$$T_{2n+\sigma(1)} = T_{n+\sigma(1)} + \tau_{2n+\sigma(1)}$$

$$UT = T_{2n+\sigma(1)}$$



Elaboración propia.

El disparo de la última transición de esta primera sub-red ($T_{2n+\sigma(1)}$) significa que se ha partido del nodo $\sigma(1)$ hacia el nodo $\sigma(2)$, de manera que la conexión entre ambas sub-redes consiste en que la primera transición de la sub-red $\sigma(2)$ debe ser igual a la última transición disparada de la sub-red $\sigma(1)$, es decir, $T_{\sigma(2)} = UT$. Situación que es análoga para cualquier paso de un nodo $\sigma(i)$ a uno $\sigma(i+1)$, por lo que la segunda parte del algoritmo de simulación de la RdP consiste en un bucle que itera hasta que se agotan las sub-redes o nodos a visitar considerando que la relación entre una sub-red y otra es a través de UT, así como se muestra en la Figura 19.

Figura 19. Segunda parte algoritmo de simulación

Para cada $i = 2$ hasta n :

$$T_{\sigma(i)} = UT$$

$$T_{n+\sigma(i)} = T_{\sigma(i)} + \tau_{n+\sigma(i)}$$

$$T_{2n+\sigma(i)} = T_{n+\sigma(i)} + \tau_{2n+\sigma(i)}$$

$$UT = T_{2n+\sigma(i)}$$

fin del ciclo

Elaboración propia.

El ciclo mostrado empieza en 2 porque para el primer nodo de la secuencia la lógica es la de la Figura 18, donde $T_{\sigma(1)}$ se iguala al parámetro del tiempo de inicio del tour T_0 . Y cuando el ciclo acaba, con el valor último de UT se obtiene la duración total del tour que es la función de desempeño que se planteó para el TDTSP.

4.2 Algoritmo genético basado en redes de Petri para la solución del TDTSP considerando el problema de inicio óptimo

En esta sección se parte de lo anteriormente planteado respecto al modelamiento del TDTSP en RdP y la respectiva simulación de la red, para diseñar la propuesta metodológica de un algoritmo que integre redes de Petri y algoritmos genéticos para la solución del problema, dando así cumplimiento al tercer objetivo específico de la tesis.

El TDTSP puede pensarse desde un punto de vista matemático, donde se obtienen formulaciones como la planteada en esta tesis (modelo (38)-(44)), con la que se puede recurrir a solvers como herramienta de resolución del modelo. Pero esta alternativa tiene limitantes respecto a la cardinalidad de los problemas, puesto que para problemas de tamaños reales la complejidad computacional

imposibilita la optimización. Por esto se hace necesario buscar alternativas como la descomposición en sub-problemas de menor complejidad, en donde el OSTP ((45)-(49)) es un claro ejemplo de ello.

Sin embargo, el problema también puede pensarse desde el paradigma de redes de Petri, que permite encontrar representaciones del problema como la planteada en la sección anterior (Figura 15), en donde el problema de secuenciación de nodos equivale a un problema de secuenciación de transiciones, que tiene como principal ventaja la estructura lógica de la red que garantiza la generación de secuencias siempre factibles, puesto que cada sub-red tiene como inicio una plaza P_i en donde un token representa que el nodo asociado no ha sido visitado, y cuando se dispara la transición de inicio de viaje hacia el nodo (T_i), dicha plaza queda vacía, de manera que nunca puede volverse a disparar esa transición, lo que significa que nunca puede volverse a visitar el nodo correspondiente.

Adicionalmente, el modelo en RdP permite usar simulación para determinar el tiempo total de un tour previamente dado, así como también dado el tiempo de inicio. Es decir, es una herramienta con la que se pueden explorar distintas alternativas de secuencias y tiempos de inicio para ir evaluando con simulación e ir buscando, con alguna lógica, mejorar las alternativas a evaluar para encontrar buenas soluciones al problema.

En esta tesis se propone que dicha lógica de búsqueda corresponda a un algoritmo genético, ya que es un metaheurístico que ha mostrado buenos desempeños en problemas de secuenciación, que es precisamente uno de los problemas a resolver en la RdP: secuenciar las primeras transiciones T_i ($i=1,2,..., n$). Al aplicar el AG a secuencias de transiciones y no directamente de nodos, se resuelve el problema de las no factibilidades comúnmente generadas por estos algoritmos

Entonces, en la producción de secuencias de transiciones es necesario garantizar la no repetición de estas, tal como lo exige la RdP, para lo cual se usa una estructura de cromosomas similar a la propuesta por Caballero & Mejía (2006) que consiste en una serie de genes que representan de manera codificada la transición a disparar, y cuya lógica de decodificación consiste en la interpretación secuencia de cada gen del cromosoma a partir de cada cual se escoge la transición a disparar del conjunto de transiciones no disparadas hasta ese momento, usando como criterio el residuo entero de la división entre el gen y la cantidad de transiciones no disparadas, más uno. Obteniendo así el valor que identifica de manera única la transición escogida del conjunto de transiciones asociadas a los nodos no visitados.

En términos prácticos supóngase que las transiciones pendientes por disparar se almacenan en un vector ordenado, entonces si el residuo entero de la división entre el gen y el tamaño de dicho vector, más uno, da como resultado k , entonces la transición escogida es la que se encuentra en la posición k del vector. Naturalmente que el tamaño del cromosoma debe corresponder con la cantidad de nodos a visitar, y los genes deben ser números enteros entre 1 y dicha cantidad de nodos a visitar. En la Tabla 2 se ilustra la interpretación de un cromosoma arbitrario para un ejemplo de 4 nodos a visitar, cuya secuencia de visitas decodificada es 1-4-3-2, como se observa en la columna “Nodo”.

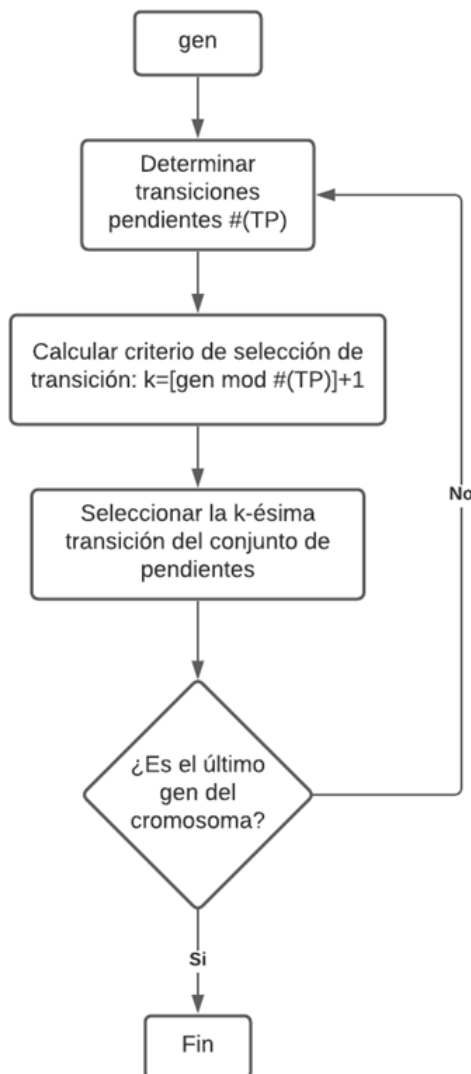
Tabla 2. Interpretación cromosoma

Cromosoma		Transiciones pendientes	Interpretación del gen	Transición escogida	Nodo
Gen 1	4	T1, T2, T3, T4	$(4 \bmod 4) + 1 = 1$	T1	1
Gen 2	2	T2, T3, T4	$(2 \bmod 3) + 1 = 3$	T4	4
Gen 3	1	T2, T3	$(1 \bmod 2) + 1 = 2$	T3	3
Gen 4	2	T2	$(2 \bmod 1) + 1 = 1$	T2	2

Elaboración propia.

Para la generalización de este método de interpretación del cromosoma, defínase TP como el conjunto de transiciones pendientes respecto a un gen determinado, entonces la cantidad de transiciones pendientes (tamaño de TP) se denota como $\#(TP)$. Por consiguiente, la k-ésima transición escogida a partir de un gen dado se calcula cómo $k = [gen \bmod \#(TP)] + 1$, y esta ecuación se aplica secuencialmente a cada gen del cromosoma para decodificarlo, de manera que el conjunto TP también va cambiando a medida que se van seleccionando transiciones. Esta dinámica se ilustra en el flujo de la Figura 20.

Figura 20. Diagrama de flujo decodificación genes

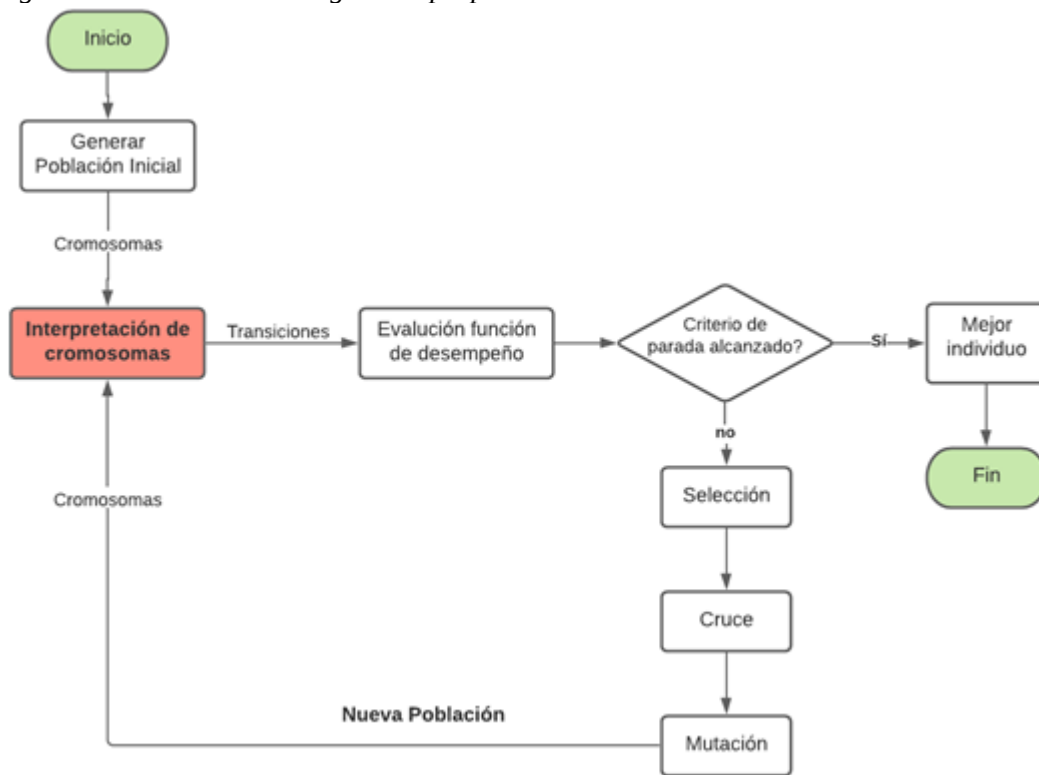


Elaboración propia.

En este punto se puede ir esbozando, entonces, el algoritmo propuesto, que se basa en la estructura general de un algoritmo genético pero que se va a ir ajustando a la aplicación en el modelo de RdP, y el primer elemento que debe considerarse es la fase adicional de interpretación de cromosomas que es en la que se generan las secuencias de transiciones correspondientes a partir de las cuales se puede evaluar la función de desempeño. En este orden de ideas, el primer avance del algoritmo propuesto se esboza en la Figura 21 donde se muestra en rojo la fase de interpretación de cromosomas en la que

se generan las secuencias de transiciones para cada cromosoma que luego entran a evaluarse y así seguir el flujo de los operadores genéticos.

Figura 21. Primer avance algoritmo propuesto

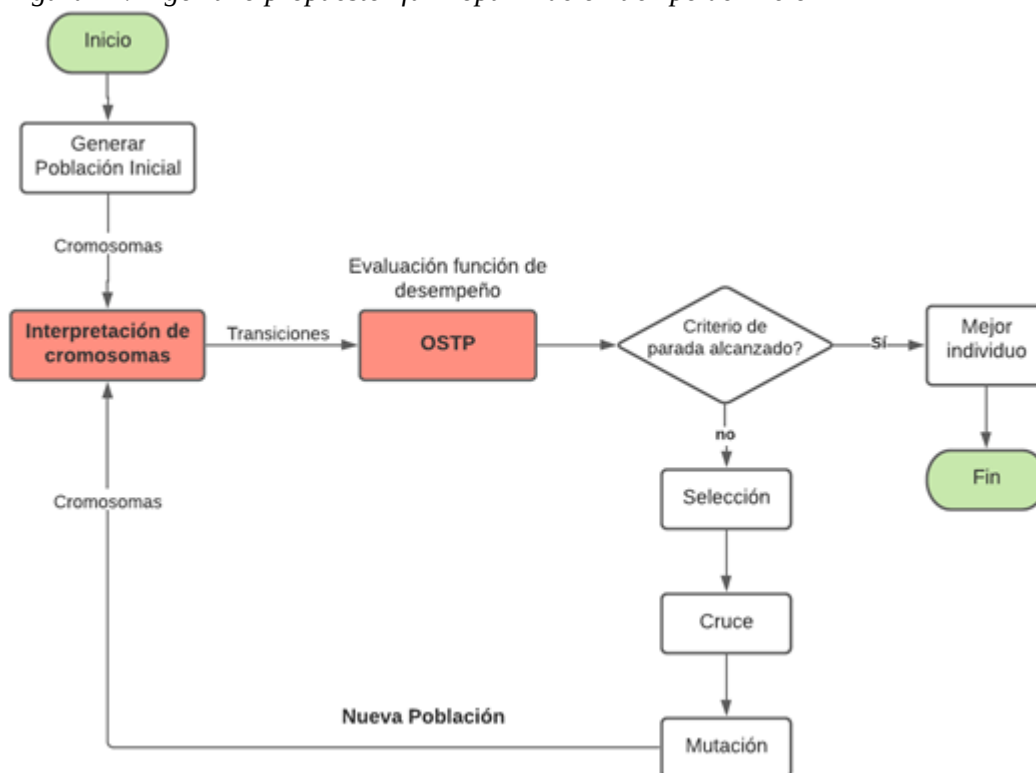


Elaboración propia.

Ahora, la secuencia de transiciones o de nodos es solo una parte del problema en el TDTSP, otro problema es el de determinar el tiempo de inicio de cada secuencia generada, el cual debe tenerse en cuenta en la fase de evaluación, ya que al ser un problema tiempo dependiente, para determinar la duración total del recorrido no basta con saber solo la secuencia, sino también la evolución temporal estricta que va determinando la duración de cada uno de los tramos del recorrido.

El problema del tiempo de inicio es de naturaleza continua, lo que hace que su exploración sea poco eficiente desde el algoritmo genético, así que se propone abordar este problema desde un modelo de programación lineal entera mixta con el que se optimice el tiempo de inicio para cada secuencia, es decir, dentro del algoritmo propuesto en la fase de evaluación incluir el modelo para el OSTP ((45)-(49)) descrito en el capítulo 2. En la Figura 22 se muestra esta propuesta en el flujo del algoritmo.

Figura 22. Algoritmo propuesto "full" optimización tiempo de inicio



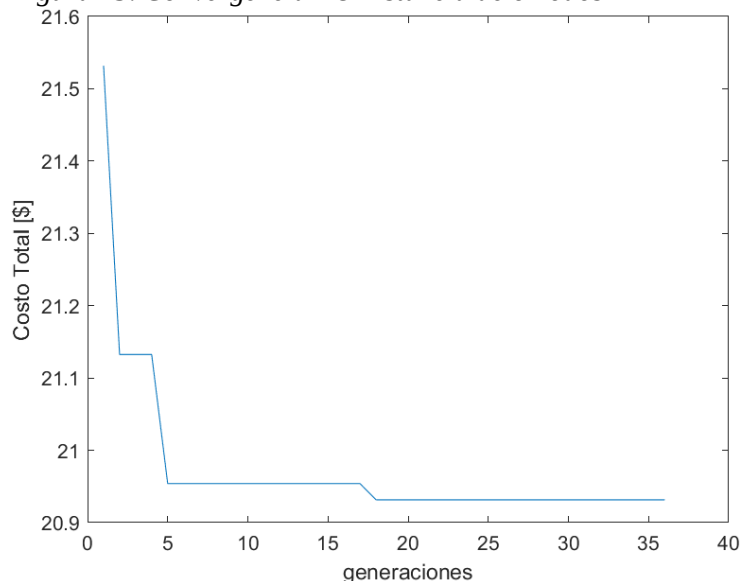
Elaboración propia.

En esta propuesta, la fase de evaluación se hace precisamente desde el modelo para el OSTP, el cual determina el tiempo de inicio que minimiza la duración total de una secuencia dada. Y de esta manera se completan todas las fases necesarias del algoritmo propuesto, el cual se procede a probar en una instancia pequeña de 6 nodos.

Para esta primera experimentación, con el objetivo de tener la solución óptima de la instancia con la que se pueda comparar el desempeño del algoritmo propuesto, se programa en el software AMPL el modelo matemático para el TDTSP descrito en el capítulo 2 ((38)-(44)) y se resuelve el modelo con los datos de la instancia usando el solver CPLEX, obteniendo como resultados una secuencia dada por {0-5-6-1-2-3-4-0}, donde 0 es el depósito. El tiempo total de recorrido óptimo fue de 20.9315 segundos, con un tiempo de inicio óptimo de 0 segundos, es decir, en el momento más temprano posible.

El algoritmo propuesto se programa en MATLAB® y se parametriza con un 50% de probabilidad de mutación, con 90 individuos de población y 90 de tamaño de población. Luego se corre para la instancia en cuestión y la secuencia de visitas está dada por {0-5-6-1-2-3-4-0}, un tiempo total de recorrido de 20.9315 segundo, con un tiempo de inicio de 0 segundos. Al comparar estos resultados con los obtenidos con la optimización en AMPL, se puede constatar que la solución del algoritmo corresponde a la solución óptima del problema, y en la Figura 23 se muestra la convergencia del algoritmo hacia dicha solución.

Figura 23. Convergencia FO instancia de 6 nodos



Elaboración propia.

A pesar del buen desempeño mostrado por el algoritmo al llegar al óptimo, se evidencia un problema en el tiempo de cómputo, que para esa de 6 nodos fue de 517.14 segundos, que es un tiempo relativamente largo para una instancia pequeña, y que se explica por el costo computacional de optimizar todos los individuos de todas las generaciones, es decir, de correr gran cantidad de veces el modelo para el OSTP. Así que resulta necesario plantear una alternativa que permita mejorar el tiempo computacional del algoritmo, especialmente pensando en la aplicación a instancias de mayor tamaño.

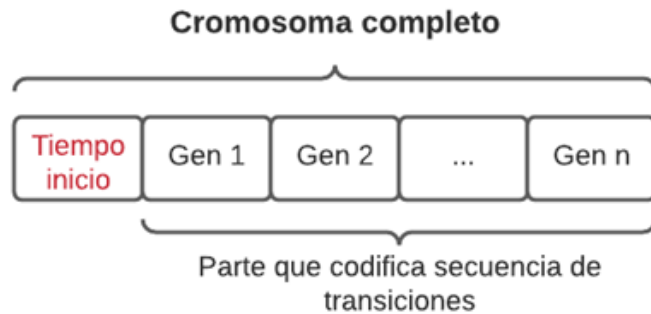
Dicha alternativa debe ser, entonces, respecto a la optimización estricta de todos los individuos en todas las generaciones, es decir, debe pensarse en otra manera de generar los tiempos de inicio de los individuos que no se vayan a optimizar, así como también la manera como éstos se van a evaluar, dada la secuencia y el tiempo de inicio.

En este orden de ideas, la propuesta inicia con definir el criterio de cuándo a una secuencia se le optimiza el tiempo de inicio y cuándo no, para lo cual se hace un símil con el concepto de *probabilidad de mutación*, que es un parámetro que aproxima el porcentaje de los individuos de la población que serán mutados. De manera que para el caso de la optimización se define también una **probabilidad de optimización** que determina el porcentaje de los individuos de la población que serán optimizados en cada generación, y de esta forma el modelo del OSTP no se correrá sobre todos los individuos de todas las generaciones sino solamente aquellos que determine la probabilidad de optimización. Cabe anotar que esta idea de definir una probabilidad de optimización es absolutamente original de este trabajo; no se encontró ninguna referencia bibliográfica en la que se plantee un concepto similar, lo cual exige que se hagan las experimentaciones respectivas para determinar la efectividad de la idea en el desempeño del algoritmo, tal como se va a hacer más adelante en este documento.

Con esta idea de la probabilidad de optimización, hay algunos individuos para los que no se va a resolver el OSTP, lo cual sugiere la necesidad de definir de qué manera se obtendrán los tiempos de inicio de sus secuencias y cómo se llevará a cabo la fase de evaluación para ellos. Respecto a la primera interrogante, se plantea seguir el principio de la *memoria genética* de los AG, en virtud de la

cual se va conservando memoria sobre las mejores características de los individuos representadas en sus genes. Así que resulta conveniente pensar en el tiempo de inicio de una secuencia, como si fuera un gen del cromosoma, ya que es una característica fundamental del individuo. En la Figura 24 se muestra, entonces, un ejemplo de la estructura de un individuo del algoritmo para un problema de n nodos, en donde se diferencia el gen del tiempo de inicio (color rojo) de los genes que codifican la secuencia de transiciones.

Figura 24. Estructura de cromosoma con gen de tiempo

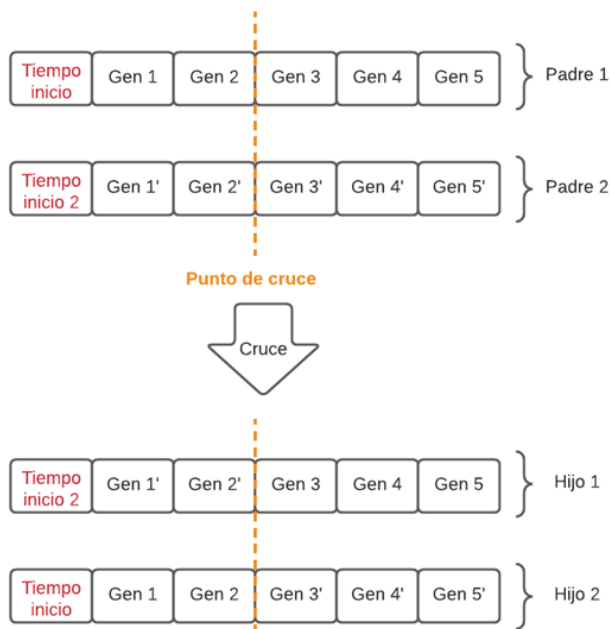


Elaboración propia.

Esta estructura del cromosoma facilita la conservación de memoria genética sobre los tiempos de inicio, especialmente porque permite incluirlos en las operaciones de cruce y mutación. En la operación de cruce el gen de tiempo entra a jugar como un gen normal, puesto que la idea fundamental del cruce es que dados dos padres, se define un punto de cruce que indica el gen sobre el cual se cortan los cromosomas padres, para luego cruzar las partes generadas de cada padre, formando así un par de hijos, lo cual, como se ilustra en la Figura 25, no implica un cambio de ubicación del gen de tiempo, sino que solamente corresponde a que el tiempo de inicio de un padre se intercambia con el del otro padre para así generar dos hijos conservando memoria genética de los tiempos.

Figura 25. Ejemplo de operación de cruce

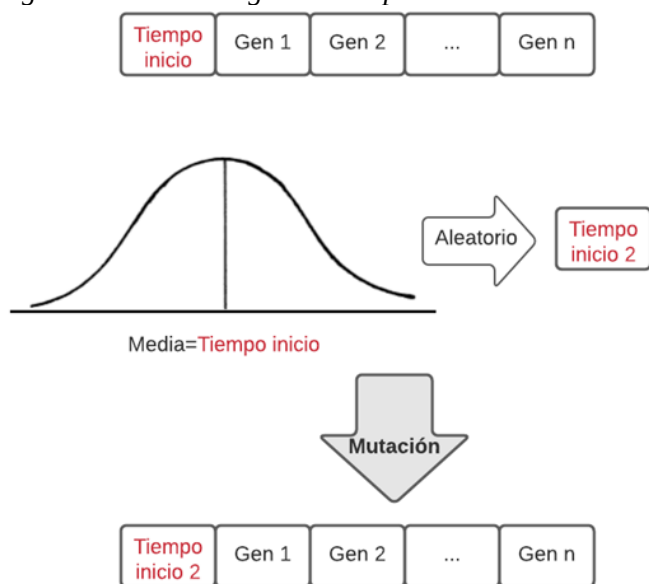
Caso $n=5$



Elaboración propia.

En contraste, para el caso de la mutación el gen de tiempo no puede considerarse como un gen normal, puesto que su naturaleza es diferente a la de los genes que codifican la secuencia de transiciones, de manera que no se puede colocar ese gen de tiempo dentro de la cadenas de genes codificadores de la secuencia, lo cual carecería de sentido. Por tal motivo se debe definir una operación de mutación diferente para este gen en particular, buscando conservar el principio de la mutación que es cambiar los valores de ciertos genes seleccionados. Por lo que una buena idea es la generación aleatoria de ese gen de tiempo a partir de una distribución normal con media igual al valor original del gen (tiempo de inicio original). Es decir, si se decide mutar cierto individuo, debe hacerse la mutación normal para la parte del cromosoma correspondiente a los genes que generan la secuencia de transiciones, y por otro lado se propone mutar el gen correspondiente al tiempo de inicio generando un número aleatorio alrededor del valor del tiempo original, para de esta manera conservar la memoria genética. En la Figura 26 se muestra la lógica de mutación para ese gen de tiempo.

Figura 26. Mutación gen de tiempo

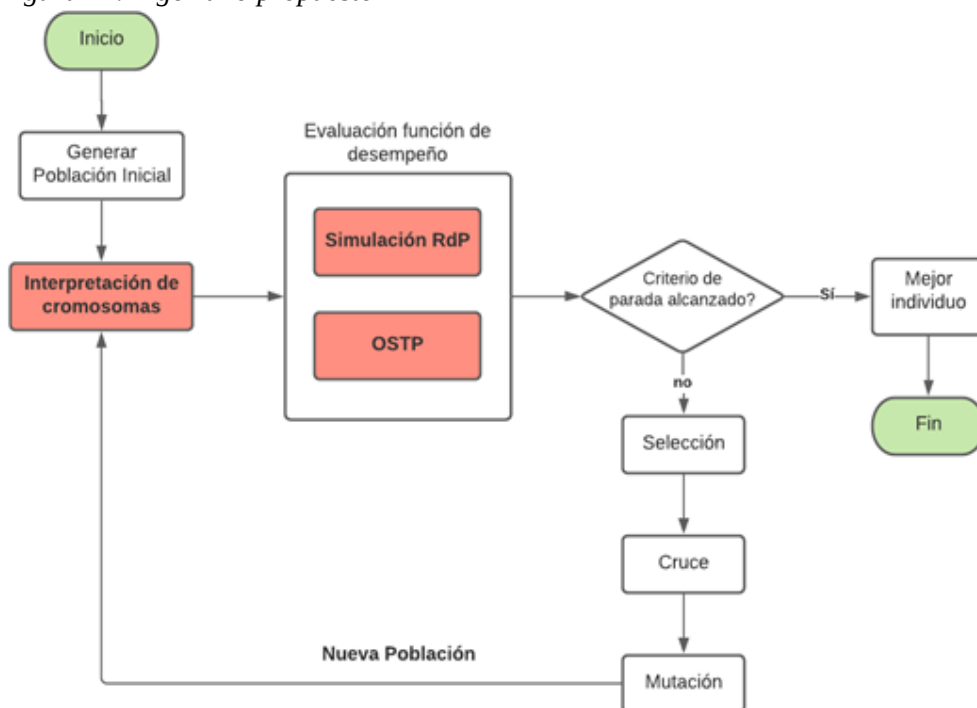


Elaboración propia.

Se ha insistido en que estas estrategias de generación de los tiempos de inicio se basan en la lógica de la memoria genética, lo cual supone que los tiempos de la población inicial son un buen punto de partida para comenzar la exploración. Por lo tanto, en esta propuesta metodológica se optimizan los tiempos de inicio para todos los individuos de la población, para así tener genes de tiempo de inicio con características deseables de conservar.

Habiendo definido la forma de generar los tiempos de inicio para los individuos que no se optimizan, la segunda interrogante corresponde a cómo realizar la evaluación de dichos individuos, respecto a lo cual la red de Petri juega un papel relevante. En la sección anterior se muestra cómo la simulación de la RdP permite, a partir de una secuencia y un tiempo de inicio, obtener el valor de la función objetivo del problema (tiempo total del tour), es decir, permite evaluar una secuencia y un tiempo de inicio. Así que en esta tesis se plantea agregar dentro de la fase de evaluación, la lógica de simulación de la RdP (Figura 18 y Figura 19) como estrategia de evaluación de los individuos que no se optimizan. En la Figura 27 se ilustra, entonces, la propuesta metodológica final para la solución del TDTSP, mostrando en rojo las etapas en las que se integra la RdP y el OSTP al algoritmo genético base.

Figura 27. Algoritmo propuesto



Elaboración propia.

Es pertinente ahora, evidenciar el impacto de la propuesta de la probabilidad de optimización en el tiempo de computo, para así validar la efectividad de esa idea. Se diseña, entonces, un marco experimental en el que se parte de la misma instancia de 6 nodos y la misma parametrización realizada en la primera experimentación del algoritmo full optimización, es decir, una probabilidad de mutación del 50%, 90 generaciones y un tamaño de población de 90 individuos. Y como lo que se desea observar es el impacto de la probabilidad de optimización, precisamente es ese parámetro el que se va a variar de manera sistemática desde 10%, hasta 90% con aumentos de 10% en cada experimento, como se muestra en la Tabla 3 donde se pueden observar también los resultados.

Tabla 3. Efecto probabilidad de optimización, instancia de 6 nodos

% optimización	Ruta óptima	Tiempo inicio óptimo [seg]	Tiempo total óptimo [seg]	Tiempo de cómputo [seg]	Generación en que converge	Generación de Stop
0.1	0-5-6-1-2-3-4-0	0	20.9315	101.68	28	37
0.2	0-5-6-1-2-3-4-0	0	20.9315	162.85	14	36
0.3	0-5-6-1-2-3-4-0	0	20.9315	227.08	29	38
0.4	0-5-6-1-2-3-4-0	0	20.9315	288.08	7	36
0.5	0-5-6-1-2-3-4-0	0	20.9315	334.17	14	36
0.6	0-5-6-1-2-3-4-0	0	20.9315	399.51	10	36
0.7	0-5-6-1-2-3-4-0	0	20.9315	465.68	15	36
0.8	0-5-6-1-2-3-4-0	0	20.9315	576.42	2	36
0.9	0-5-6-1-2-3-4-0	0	20.9315	618.99	7	36

Elaboración propia.

Es evidente como a medida que se aumenta la probabilidad de optimización, el tiempo de computo también lo hace, pero esto sin impactar la calidad de las soluciones, que se mantienen en el óptimo

para todos los escenarios de dicha probabilidad. De manera que se logra el objetivo de disminuir el tiempo de computo del algoritmo, que comparando el escenario de 10% de optimización con el de 90%, se tiene que el tiempo de computo aumenta de 101.68 a 618.99 segundos, respectivamente, es decir, la disminución en cuanto a tiempo computacional corriendo el algoritmo con un 10% de optimización es del 83.57% respecto a la corrida con 90% de optimización. Por tanto, se puede concluir la pertinencia de la propuesta metodológica diseñada en esta tesis.

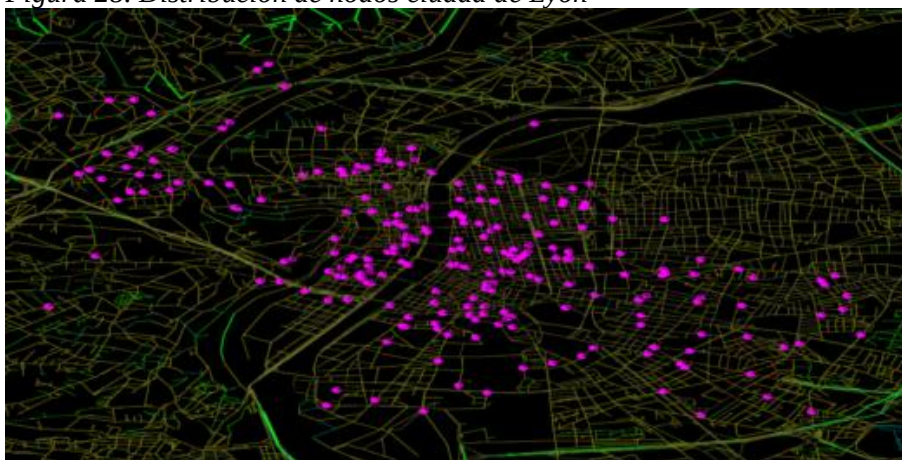
4.3 Validación de la metodología en instancias del problema

Una vez definida la propuesta metodológica del algoritmo basado en RdP y AG, en éste capítulo se prueba dicha metodología en instancias de distintos tamaños para validar el comportamiento y desempeño del algoritmo, dando así cumplimiento a lo planteado en el cuarto objetivo específico de la tesis.

El TDTSP se aplica a varios problemas reales de programación, dentro de los que se encuentran las entregas en contextos urbanos, donde los tiempos de viaje varían considerablemente en función del momento del día, debido a las condiciones de congestión vial en ciertas horas pico. Bajo esta premisa es que Melgarejo, Laborie, & Solnon (2015) proponen un conjunto de instancias del problema, basadas en datos reales de tráfico de la ciudad de Lyon (Francia) en donde se evidencia la naturaleza tiempo dependiente en los viajes.

Para la construcción de las instancias los autores antes citados han accedido a datos históricos de 6 años, capturados por 630 sensores ubicados en las principales vías de la ciudad de Lyon, los cuales miden la velocidad de los vehículos. A partir de dichos datos, generaron un modelo predictivo que estima las velocidades para ciertos intervalos de tiempo del día. Y con estas estimaciones se construye la matriz de tiempos para 255 nodos seleccionados aleatoriamente de una lista de direcciones de puntos de entrega de los transportadores de la ciudad, cuya distribución se observa en la Figura 28.

Figura 28. Distribución de nodos ciudad de Lyon



Fuente: (Melgarejo et al., 2015).

La matriz de tiempos de viaje corresponde a una función stepwise que contiene información para los arcos entre los 255 nodos, y a partir de la cual se pueden generar las instancias particulares propuestas por los autores que pueden ser de 10, 20, 30, 50 o 100 nodos y que se pueden descargar de la siguiente página: <http://perso.citi-lab.fr/csolnon/TDTSP.html>. Los archivos de cada instancia se componen de un listado de la cantidad de nodos correspondiente, en donde la primera columna del listado contiene

el índice de cada nodo, el cual debe buscarse en la matriz de tiempos, y en la segunda columna el tiempo de servicio en el nodo. Es decir, las instancias se construyen cruzando el archivo del listado de los nodos de la instancia, con la matriz que contiene la información de todos los nodos.

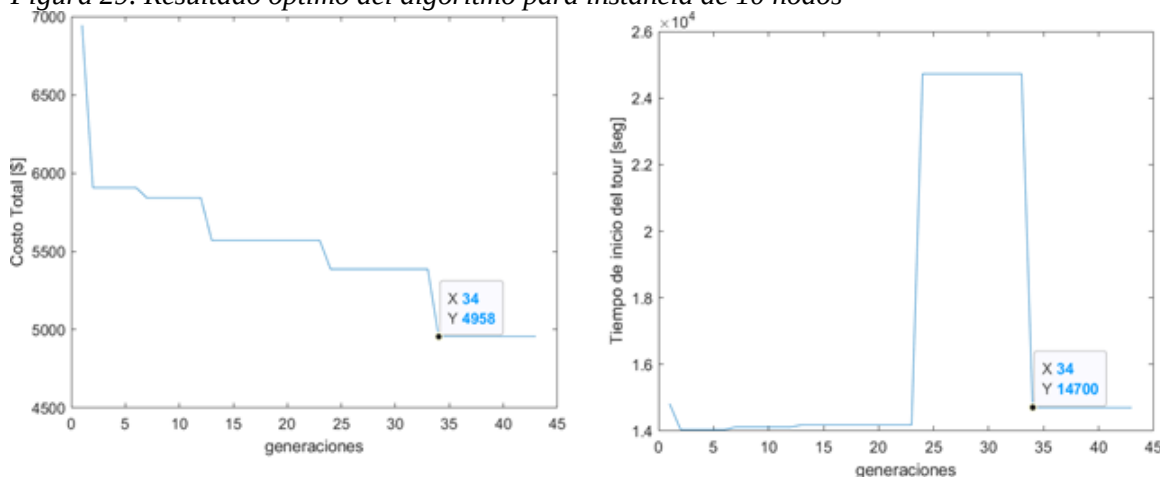
Teniendo en cuenta lo anterior, para validar el algoritmo propuesto en esta tesis se toman tres instancias: de 10, 50 y 100 nodos; considerando 10 intervalos de tiempos para las tres. Los resultados computacionales se muestran en el siguiente apartado.

Resultados de cómputo en instancias

Para la instancia de 10 nodos y 10 intervalos de tiempo, como primera medida se prueba el modelo matemático del problema, que como se mencionó anteriormente, se ha programado en AMPL. Al cual se le ingresan los datos de la instancia y se resuelve con el solver CPLEX, obteniéndose como solución óptima una ruta dada por {0-1-7-5-9-8-2-3-4-6-0}, que con un tiempo de inicio óptimo de 14697.5 segundos, da un tiempo total de recorrido óptimo de 4957.54 segundos.

Por otro lado, el algoritmo propuesto en esta tesis se parametriza con un tamaño de población de 200 individuos, 90 generaciones, un porcentaje de elitismo del 5% y una probabilidad de mutación de 50%. Y en primera instancia se experimenta con una probabilidad de optimización del 100%, es decir, optimizando el tiempo de inicio de todos los individuos de todas las generaciones, caso en el que el algoritmo llega a la siguiente ruta: {0-1-7-5-9-8-2-3-4-6-0}, a un tiempo de inicio de 14697.5 segundos y a un tiempo total de recorrido de 4957.54 segundos, como se muestra en la Figura 29 en donde la gráfica izquierda muestra la convergencia de la función de desempeño a medida que transcurren las generaciones (valor del mejor individuo de cada generación), y el gráfico derecha se muestra el comportamiento de los tiempos de inicio de los mejores individuos de las generaciones. Cabe resaltar que aunque se parametrizaron 90 generaciones, el algoritmo para en la generación número 43 (generación de stop), puesto que allí se cumple el criterio de parada por convergencia.

Figura 29. Resultado óptimo del algoritmo para instancia de 10 nodos



Elaboración propia.

En estos resultados se observa como el algoritmo llega a la solución óptima del problema, pero lo hace en un tiempo de cómputo de 5800 segundos que, como se discutió en la instancia de 6 nodos, es un tiempo mejorable a través de variaciones en el porcentaje de optimización. Y dado que para esta instancia de 10 nodos se logró obtener la solución óptima como punto de referencia, se propone

realizar una experimentación análoga a la que se hizo para la instancia de 6 nodos (Tabla 3), en donde se varíen los porcentajes de optimización, para lograr evidenciar el impacto en el desempeño del algoritmo. Se parte, entonces, de la parametrización inicial del algoritmo y se generan escenarios del porcentaje de optimización con avances de 0.1, partiendo desde 0.1 hasta 0.9, y obteniendo para cada caso las soluciones mostradas en la Tabla 4.

Tabla 4. Efecto probabilidad de optimización, instancia de 10 nodos

% optimización	Ruta óptima	Tiempo inicio óptimo [seg]	Tiempo total óptimo [seg]	Tiempo de cómputo [seg]	Generación en que converge	Generación de Stop
0.1	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	1083	47	56
0.2	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	1818	38	47
0.3	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	2456	41	50
0.4	0-3-4-6-1-7-5-9-8-2-0	24726.8	5386.38	2177	16	36
0.5	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	2823	31	40
0.6	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	4433	40	49
0.7	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	4139	19	36
0.8	0-6-1-7-5-9-8-2-3-4-0	25924.2	5052.15	5561	38	47
0.9	0-1-7-5-9-8-2-3-4-6-0	14697.5	4957.54	5200	35	44

Elaboración propia.

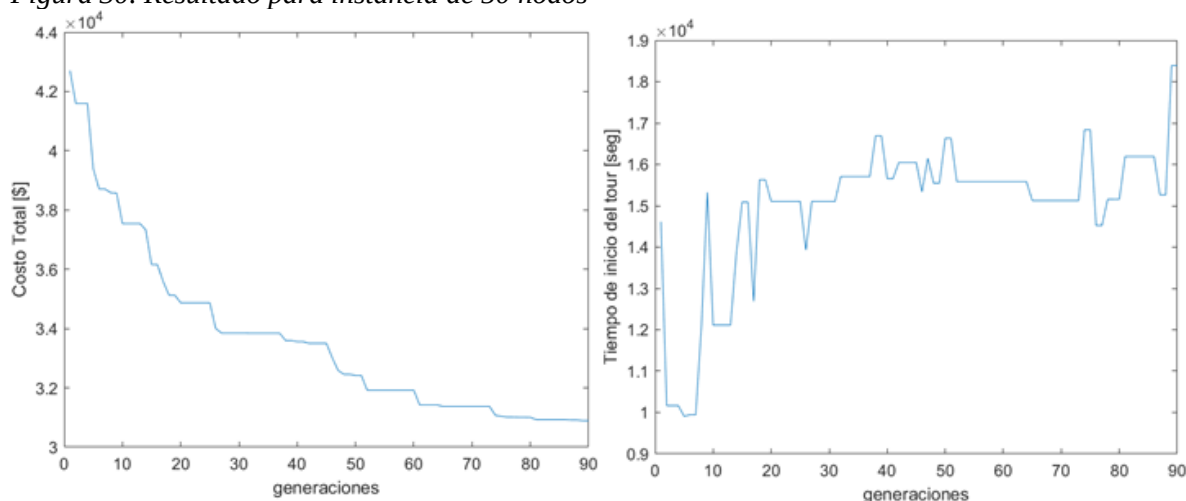
Estos resultados muestran que únicamente en dos escenarios no se llega a la solución óptima. Para la probabilidad de optimización del 40% el valor obtenido de la función objetivo fue de 5368.38, y para el caso del 80% el valor fue de 5052.15, de resto en todos los otros escenarios se evidencia la solución óptima de la instancia. Estos dos casos no marcan un comportamiento sistemático de empeoramiento de la calidad de soluciones de algoritmo, sino que responden al componente aleatorio que introduce la generación inicial del algoritmo, la cual es generada aleatoriamente en cada escenario, lo que hace que en cada escenario el algoritmo corra desde distintos puntos de partidas, haciendo posible la ocurrencia de estos dos casos en los que no se obtiene la solución óptima del problema, pero se llega a muy buenas soluciones cerca del óptimo: para el caso del 40% de optimización el GAP es del 8.29%, y para el caso del 80% de optimización el GAP es de tan solo el 1.91%.

Adicionalmente, se observa cómo el aumento en los porcentajes de optimización induce un aumento sistemático de los tiempos de cómputo del algoritmo, debido a que a mayores porcentajes, con mayor frecuencia el algoritmo debe entrar a resolver el modelo matemático del OSTP, lo cual tiene un costo computacional mayor a la otra opción de evaluación que es correr la simulación de la RdP del problema. Sin embargo, se puede observar que para los casos con menores porcentajes de optimización el algoritmo tiene un mejor desempeño computacional conservando la calidad de las soluciones, tal como se había concluido en el análisis de las experimentaciones con la instancia de 6 nodos.

Ahora, debido al buen desempeño que el algoritmo ha mostrado con 10% de probabilidad de optimización, se decide usar este valor para la parametrización con la que se corren las instancias más grandes de 50 y 100 nodos, y que se completa con los siguientes valores: 200 de tamaño de población, 90 generaciones, 5% de elitismo y 50% de probabilidad de optimización. Cabe anotar que para las instancias de 50 y 100 nodos la complejidad computacional imposibilita la resolución del problema de manera óptima con el modelo matemático, por lo que el algoritmo propuesto toma especial relevancia práctica.

En la Figura 30 se observan los resultados para la instancia de 50 nodos, en la gráfica de la izquierda se muestra el comportamiento de la función de desempeño y cómo el algoritmo la mejora a medida que transcurren las generaciones, hasta llegar a un valor de 30891.08 segundos, en la gráfica de la derecha se observa el comportamiento del tiempo de inicio, el cual tiene un valor de 18389.61

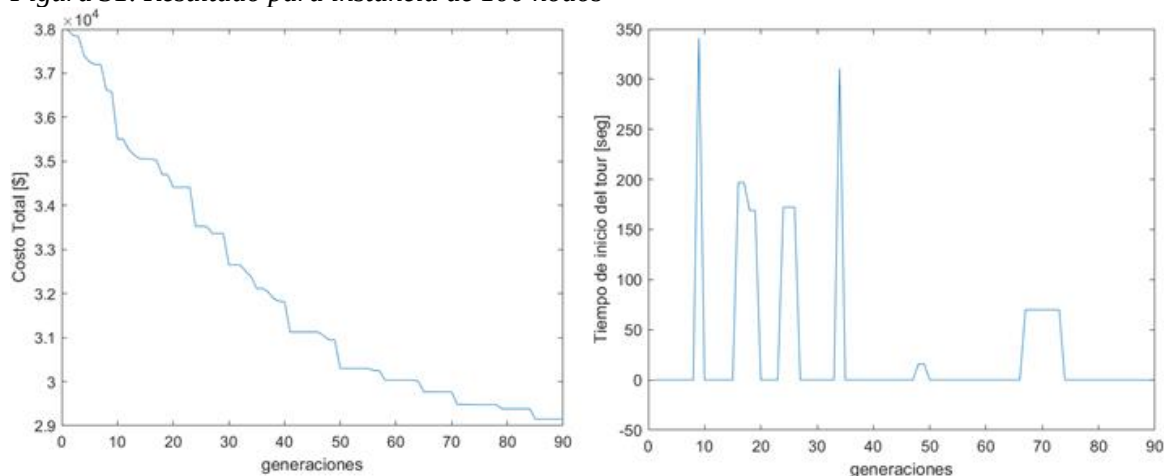
Figura 30. Resultado para instancia de 50 nodos



Elaboración propia.

Para la instancia de 100 nodos la parametrización del algoritmo es la misma que para la de 50 nodos, y en la Figura 31 se muestran los resultados, en donde se observa que el valor de la función objetivo obtenida al final de las 90 generaciones es de 29151 segundos, y un tiempo de inicio de 0 segundos, es decir, partir en el momento más temprano posible.

Figura 31. Resultado para instancia de 100 nodos



Elaboración propia.

En síntesis, se han argumentado las ventajas teóricas que sustentan la idea de un algoritmo que integre RdP y AG como metodología de solución para el TDTSP, se describe la propuesta realizada para dicho algoritmo y se valida el buen desempeño obtenido en distintas instancias del problema, llegando a las soluciones óptimas para los casos en los que se puede tener la comparación con el modelo matemático, y mostrando buenos desempeños en el mejoramiento de las soluciones con la evolución genética de generación en generación.

5. APLICACIÓN A CASO REAL EN PROCESO DE RECOGIDA Y VENTA DE LECHE EN UN MUNICIPIO DE COLOMBIA

Dada la pertinencia del algoritmo propuesto en el capítulo anterior para la solución de problemas tiempo dependientes, en el presente capítulo se muestra una aplicación a un caso real donde la dependencia temporal es respecto a tiempos de servicio y penalizaciones por pérdida de ventas, que son elementos importantes en los procesos de recogida de ítems perecederos con descargue en depósito para la venta. Se presenta un caso real para la recolección y venta de leche en un municipio de Colombia, que el autor y los directores de la presente tesis describen detalladamente y resuelven a través de programación matemática en (Osorio et al., 2021).

En la práctica es común que los problemas de ruteo revistan mayor complejidad respecto al planteamiento básico del TSP, como ocurre con las consideraciones de ventanas de tiempo (TSPTW) o los tiempos de servicio tiempo dependiente reconocido por sus siglas en inglés como TSP-TS y que será objeto de análisis en este capítulo. Cada problema particular tiene un marco con elementos propios cuya relevancia es necesario considerar en cualquier esfuerzo de abstracción, que requiere involucrar las particularidades del problema en busca de un alto grado de representatividad.

Para el caso abordado en este capítulo sobre la operación de recogidas refrigeradas, se consideran el comportamiento de la pérdida de ventas y los tiempos de descargue, como elementos tiempo dependientes modelados con funciones piecewise (por partes). Además se incluye el costo de espera de la unidad de refrigeración, dotando al caso con una completitud que no había sido abordada en la literatura científica hasta el trabajo de Osorio, Bravo-Bastidas, & Orejuela (2021).

Los tiempos de servicio son un elemento poco abordado en la literatura respecto del TSP, dado que el interés se ha centrado en la minimización de la distancia total recorrida o de la duración total del tour, esta última considerada únicamente como el tiempo de los recorridos, o como la suma de dichos tiempos con los tiempos de servicio fijos en cada nodo. Esa naturaleza constante de los tiempos de servicio no requiere que ellos se involucren directamente en los procesos de optimización, sino que basta con que al tiempo total de recorrido mínimo obtenido, se le sume esa cantidad constante asociada con dichos tiempos. Fue solo hasta el 2016 que Taş, Gendreau, Jabali, & Laporte hacen la primera propuesta en la literatura en la que se plantea genéricamente el TSP con tiempos de servicio tiempo dependientes (TSP-TS) desarrollando un modelo de programación entera mixta para resolverlo, en el cual sí es necesario involucrar los tiempos de servicio en cada nodo como una función del tiempo de llegada al nodo, ya que el TSP-TS busca minimizar el tiempo total definido como la suma del tiempo de los viajes y los tiempos de servicio.

Según Cacchiani, Contreras-Bolton, & Toth (2020) en muchas situaciones prácticas del TSP la consideración del tiempo de servicio tiempo dependiente es relevante, ya que factores como el personal disponible para el cargue/descargue de vehículos o la dificultad de acceso a algunos nodos en determinadas horas del día, hacen que los tiempos de servicios difieran de ciertas horas a otras, invalidando el supuesto de tiempos constantes y aportando complejidad al problema. Según Taş, Gendreau, Jabali, & Laporte (2016), la consideración de la dependencia temporal conlleva a la definición de una función de tiempos de servicio que generalmente es no lineal.

Por otro lado, según Ballou (2004) la pérdida de ventas es una variable importante a considerar en la estructura de costos de la operación logística de suministro de productos. Para Mukhopadhyay &

Goswami (2016) la pérdida de ventas conlleva a muchas empresas a grandes pérdidas no solamente monetarias, sino intangibles como la reputación, realizándose grandes esfuerzos operativos y de capital para mitigar los stock-outs o eventos de falta de inventario. Es por esta importancia del costo de pérdida de ventas y el importante esfuerzo requerido para su reducción, que Ballou (2004) menciona la necesidad de determinar el equilibrio óptimo entre ese costo y otros costos de logística como de inventario y transporte, puesto que entre estos ocurren conflictos en donde una reducción en las ventas perdidas requiere un aumento en los costos de inventario y/o de transporte, siendo en muchos casos muy costoso tener un nivel de servicio del 100% o muy cercano a éste. Sobre la búsqueda de ese equilibrio óptimo respecto a problemas de ruteo en cadenas de suministro, la literatura muestra modelos que involucran una penalización por pérdida de ventas (Giesen, Mahmassani, & Jaillet, 2005, 2010; Moosavi & Nikfarjam, 2019).

5.1 Problema abordado

El problema abordado consiste en el ruteo para la recogida de productos refrigerados sobre un conjunto de nodos dispersos geográficamente. Dicho ruteo se aborda desde el TSP, es decir, se considera un único vehículo que parte de un depósito hacia ciertos nodos, visitándolos a todos una única vez para realizar las recogidas y regresando de nuevo al depósito donde se requieren comercializar los productos recolectados. Este tipo de escenarios surgen, por ejemplo, en negociaciones en las que el comprador se encarga de recoger los productos en donde el proveedor los almacena, que en términos de las reglas de Incoterms, que rigen las operaciones de compraventa tanto nacionales como internacionales, se conoce como “En Fábrica” o en inglés “Ex Works” (EXW).

En el contexto de los problemas de recogidas, algunos depósitos tienen horarios de operación donde se realiza la venta del producto previamente recogido, lo cual define una ventana de tiempo relevante. Esta empieza con la hora de inicio de descargue de vehículos y termina con la hora en la que se cierra el establecimiento. Cuando los vehículos recolectores llegan de manera anticipada, es decir, llegan antes de la hora de inicio de descargue (inicio de la ventana), se incurre en una espera que implica el funcionamiento de la unidad de refrigeración del vehículo, generando costos por consumo de combustible que impactan la eficiencia de la operación y exigiendo por ello una mayor sincronía en el proceso.

Respecto a la mencionada ventana de tiempo del depósito, en la práctica es muy común observar que en las vecindades de la hora de inicio de descargue suele ocurrir una alta congestión de vehículos que esperan ser atendidos para comercializar sus productos, y por esa alta congestión el tiempo de servicio a los vehículos en esa franja horaria es mayor que en otros momentos, situación que sugiere considerar la dependencia del tiempo de servicio en el depósito (descargue de productos) en función del momento de arribo a él.

Adicionalmente a la hora de inicio de descargue, en este tipo de depósitos de comercialización se define una hora de inicio de ventas o de comercialización, que es el momento de apertura a los clientes, y ello resalta la ocurrencia de ciertas pérdidas de ventas en función de la tardanza del descargue del vehículo frente a esa hora de inicio de comercialización en el depósito. Además, es claro que el descargue de un vehículo después del momento de cierre del depósito implicaría una pérdida de ventas máxima, ya que en esas horas el depósito no opera, es decir, no hay comercio. Con esa misma idea, la pérdida de ventas es nula cuando el producto se descarga en el momento de inicio de la comercialización o antes, puesto que en esas horas el producto logra estar disponible desde el momento en que la demanda inicia. Pero si el producto no llega en dicho inicio o antes, a medida que transcurre el tiempo se va dejando de atender demanda potencial, incurriendo en pérdida de ventas.

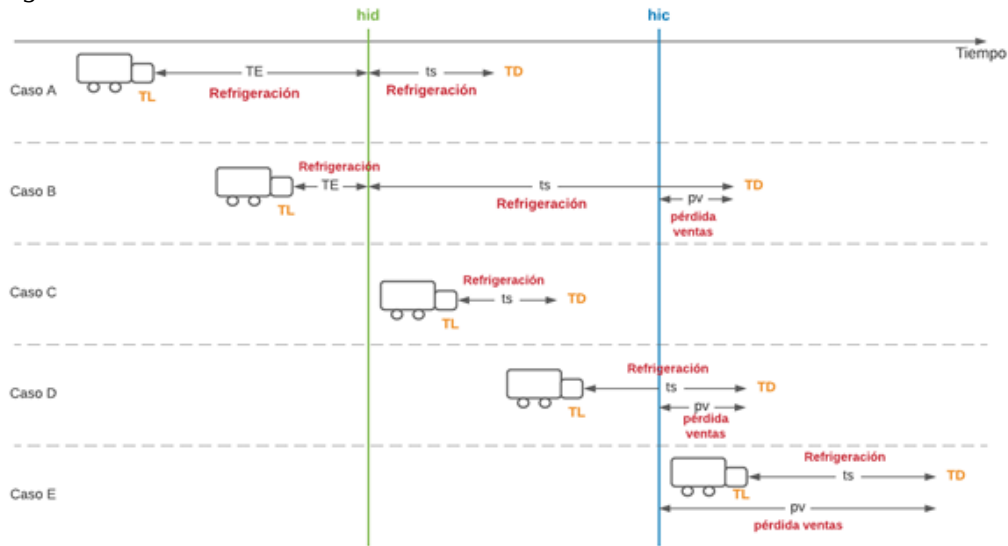
En tal sentido, el problema abordado en este capítulo consiste en un TSP de recogidas de productos refrigerados para su comercialización una vez se llegue al depósito, considerando los tres elementos discutidos anteriormente: tiempos de espera, tiempos de servicio (tiempo de descargue) tiempo dependientes y penalización por pérdida de ventas, los cuales serán componentes del costo total que también incluye el costo por consumo de combustible en el desplazamiento entre los nodos proveedores. El objetivo será entonces la minimización de ese costo total sobre el cual, el funcionamiento de la unidad de frío tiene un impacto significativo, toda vez que es un factor de costo relevante cuando los arribos se dan antes de la hora de inicio de descargues, y durante el tiempo de servicio. Por otro lado, se asume que se incurre en un costo por pérdida de ventas solo cuando la finalización del descargue del vehículo ocurre después de la hora de inicio de la comercialización de productos.

La Figura 32 ilustra el funcionamiento de los costos asociados a la espera, los tiempos de servicio y la pérdida de ventas, en donde la línea verde representa la hora en la que se inician los descargues de los vehículos en el depósito, que se denota como “*hid*”, la línea azul representa la hora en la que se inicia la comercialización de los productos, denotada como “*hic*”, y la ubicación de los camiones en dicha figura simboliza el momento de arribo al depósito (“*TL*”) en cada uno de los cinco escenarios presentados.

En el caso A el vehículo arriba antes de que se inicien los descargues de vehículos (línea verde), y en función de ese momento de llegada se le asocia un tiempo de servicio “*ts*” el cual empieza a transcurrir después de la espera “*TE*”, es decir, una vez se llegue a la “*hid*”; en términos de costos se tiene que sobre el tiempo “*TE*” se incurre en refrigeración, así mismo en el tiempo de descargue “*ts*”, y el costo por pérdida de ventas ocurrirá solamente si la finalización del tiempo de servicio o de descargue (“*TD*”) ocurre después de la “*hic*” (línea azul), que para el caso A no se tiene esa situación, como se puede observar en la Figura 32.

El caso B es un caso similar al A, en el que el arribo del vehículo se da antes de la “*hid*” pero además de los costos de refrigeración asociados al “*TE*” y el “*ts*”, se incurre en un costo por pérdida de ventas “*pv*” debido a que el “*TD*” es posterior a la “*hic*”. El caso C y D son situaciones en que el arribo (“*TL*”) ocurre después de la “*hid*” pero antes de la “*hic*”, y muestran las opciones de que el “*TD*” ocurra antes de la “*hic*” (caso C) o después (caso D), teniendo en el primero solamente costos de refrigeración sobre el “*ts*” y en el segundo, además de los costos de refrigeración, los costos por pérdida de ventas. Por último, la Figura 32 muestra el caso E en el que el arribo se da después de la línea azul, por lo que tiene costos asociados al tiempo de servicio asignado en función de su hora de llegada y a la pérdida de ventas.

Figura 32. Costos del caso

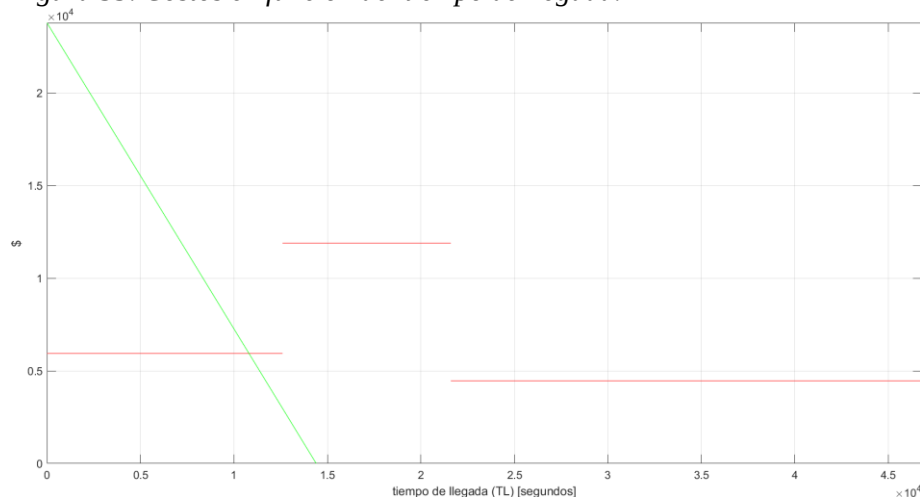


Fuente: (Osorio et al., 2021).

La descripción del problema en la Figura 32 contiene una aproximación a la naturaleza de los elementos de costo que componen la función de desempeño planteada en este trabajo, específicamente sobre los componentes de costo por espera, penalización por tiempo de servicio y penalización por pérdida de ventas. Respecto a la espera, el costo es función del tiempo de llegada al depósito (TL), puesto que si el arribo ocurre antes de la hora de inicio de descargue (hid), entonces el tiempo de espera se define como $hid - TL$, siendo el costo directamente proporcional a ese tiempo de espera, ya que es la multiplicación de este por el consumo de la unidad de frío en términos monetarios, de manera que, matemáticamente el costo por espera es una función lineal dependiente de TL con pendiente negativa que llega a cero en el tiempo igual a “ hid ” tal como se representa en la recta verde de la Figura 33.

Por otro lado, la descripción del problema sugiere que la penalización por tiempo de servicio también es una función del tiempo de llegada (TL), sin embargo, el comportamiento de ésta no es lineal como en el caso anterior, sino que corresponde a una función por partes en donde a unos determinados rangos del tiempo de llegada se le asocia un tiempo de servicio que se monetiza multiplicándose por el consumo de la unidad de frío, resultando la función representada en rojo en la Figura 33.

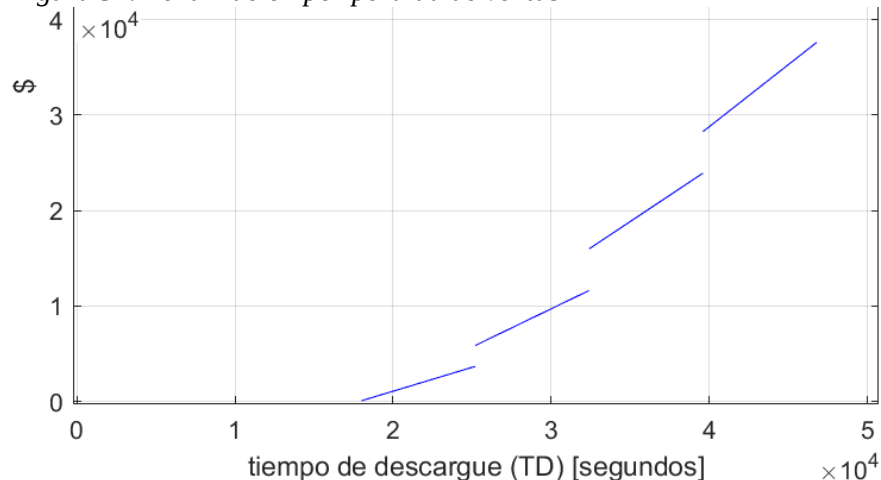
Figura 33. Costos en función del tiempo de llegada.



Fuente: (Osorio et al., 2021).

La interacción de las funciones mostradas en la Figura 33, así como la no linealidad que se evidencia en ella, son una complejidad que se acentúa aún más con la consideración adicional de la penalización por pérdida de ventas antes mencionada, y cuya naturaleza tampoco es lineal sino que consiste en una función por partes dependiente del tiempo de finalización del descargue (TD). Esto se debe a que únicamente en el instante en que se finaliza el descargue es cuando se tiene disponibilidad de los productos para su comercialización y, como se muestra en la Figura 32, si el TD ocurre después de la hora de inicio de comercialización (hic), entonces la tardanza se define como $TD-hic$ y en función de esta tardanza es que se dan las ventas perdidas en una relación directa. En la Figura 34 se observa que la función tiene como dominio el intervalo entre la hora de inicio de la comercialización y la hora de cierre del depósito y que a medida que el tiempo de descargue crece, también lo hace la penalización en segmentos de recta que definen a la función por partes, tal que la pendiente de los segmentos crece a medida que los intervalos corresponden a mayores tiempos.

Figura 34. Penalización por pérdida de ventas



Fuente: (Osorio et al., 2021).

5.2. Metodología

El problema descrito en la sección anterior consiste en determinar tanto la secuencia de nodos de recolección a visitar, como el momento de inicio del tour, tal que los costos totales incurridos sean mínimos. Como herramienta metodológica de solución se propone partir del algoritmo desarrollado en esta tesis y descrito en el capítulo anterior, el cual es necesario ajustar considerando los elementos particulares de este problema, cuya consideración conjunta recientemente se aborda en el artículo de Osorio et al. (2021), como lo son la función piecewise de pérdida de ventas y de tiempos de descarga, tiempo dependientes. Y la función para el costo de espera de la unidad de refrigeración.

Estas particularidades definen una nueva función de desempeño que es necesario reflejar en la etapa de evaluación de algoritmo. Es decir, el problema considerado tiene unas nuevas dinámicas que sugieren ajustar tanto el modelo para el OSTP, como la estructura en la que se usa la simulación de la RdP.

Al tratarse de un problema en el que los tiempos de viaje no son tiempo dependientes (la dependencia temporal corresponde a otros elementos), la duración de una secuencia cualquiera es siempre la misma, sin importar el tiempo de inicio del tour. Este hecho le da mayor protagonismo al uso de la simulación de la RdP en la etapa de evaluación del problema, puesto que de dicha simulación se obtiene la duración del recorrido de la secuencia considerando siempre que el tiempo de inicio del tour es el momento cero ($T_0=0$). Y esa duración del recorrido es el parámetro requerido tanto para el OSTP, como para el cálculo de la función de desempeño cuando no se optimiza el tiempo de inicio.

Se va a presentar primero el modelo para el OSTP, ya que su entendimiento favorece la posterior comprensión de la lógica de evaluación para los individuos no optimizados. A continuación, entonces, el modelo en el que la estructura de la función objetivo se compone de las funciones representadas en la Figura 33 y Figura 34, además del consumo de combustible asociado al recorrido. En este modelo se observa que la secuencia no es una decisión a tomar, sino que de ella lo que interesa es el parámetro de duración del recorrido, denotado como dr .

Conjuntos:

Intervalos de tiempos de servicio $[K] = \{1, 2, \dots, n\}$, índice $\rightarrow k$

Intervalos de pérdida de ventas $[H] = \{1, 2, \dots, m\}$, índice $\rightarrow h$

Parámetros:

$dr \equiv$ Duración del recorrido.

$liits_k \equiv$ Límite inferior del intervalo de tiempo de servicio k . [seg]

$lsits_k \equiv$ Límite superior del intervalo de tiempo de servicio k . [seg]

$liipv_k \equiv$ Límite inferior del intervalo de pérdida de ventas h . [seg]

$lsipv_k \equiv$ Límite superior del intervalo de pérdida de ventas h . [seg]

$cguf \equiv$ Consumo de gasolina de la unidad de frío. $\left[\frac{\text{litros}}{\text{seg}} \right]$

$cgr \equiv$ Consumo de gasolina por recorrido. $\left[\frac{\text{litros}}{\text{seg}} \right]$

$cdg \equiv$ Costo de gasolina. $\left[\frac{\$}{\text{litro}} \right]$

$ppv_h \equiv$ Penalización fija por pérdida de ventas asociada al intervalo h .

$pend_h \equiv$ Pendiente de la función de pérdida de ventas en el intervalo h

$tstd_k \equiv$ Tiempo de servicio en el depósito en el intervalo k [seg]

$hid \equiv$ Hora de inicio de descargas [seg]

Variables:

T_0 = Tiempo de partida del depósito (inicio del tour)

TD = Tiempo en el que el vehículo es descargado en el depósito

$TAID$ = Tardanza respecto a hora de inicio de descargue (hid)

$ANID$ = Anticipo o tiempo de espera respecto a hora de inicio de descargue (hid)

$TAIC$ = Tardanza respecto al inicio de la comercialización en el depósito

$ANIC$ = Anticipo respecto al inicio de la comercialización en el depósito

$TAICH_h$ = Variable auxiliar para asociar la dimensión de $TAIC$ a un intervalo h

$B_h = \begin{cases} 1, & \text{si } TAIC \in \text{intervalo } h \\ 0, & \text{si no} \end{cases}$

$Y_k = \begin{cases} 1, & \text{si } TL \in \text{intervalo } k \\ 0, & \text{si no} \end{cases}$

Función objetivo:

$$\text{Min } f = cgr * cdg * dr + cguf * cdg \left[dr + ANID + \sum_{k \in K} Y_k * tstd_k \right] + \sum_{h \in H} (TAICH_h * pend_h + B_h * ppv_h) \quad (50)$$

Restricciones:

$$(T_0 + dr) - hid = TAID - ANID \quad (51)$$

$$TD = hid + TAID + \sum_{k \in K} Y_k * tstd_k \quad (52)$$

$$TD - liipv_1 = TAIC - ANIC \quad (53)$$

$$TAIC = \sum_{h \in H} TAICH_h \quad (54)$$

$$B_h(liipv_h - liipv_1) \leq TAICH_h; \quad \forall h \in H \quad (55)$$

$$B_h(lsipv_h - liipv_1) \geq TAICH_h; \quad \forall h \in H \quad (56)$$

$$\sum_{h \in H} B_h \leq 1 \quad (57)$$

$$\sum_{k \in K} Y_k * liits_k \leq (T_0 + DR) \quad (58)$$

$$\sum_{k \in K} Y_k * lsits_k \geq (T_0 + DR) \quad (59)$$

$$\sum_{k \in K} Y_k \leq 1 \quad (60)$$

$$TD \leq lsipv_l \quad (61)$$

El primer término de la función objetivo (50) corresponde a los costos por consumo de gasolina durante el tiempo del recorrido de los vehículos, el segundo término se asocia al costo por consumo de gasolina de la unidad de refrigerado, la cual trabaja durante el recorrido (dr), además del tiempo de espera por el anticipo respecto al inicio del descargue representado por la variable $ANID$ (en la Figura 32 se muestra cómo TE), y durante el tiempo de descargue del nodo que depende del intervalo h en que el vehículo arribó (ver Figura 32); el tercer término corresponde a la penalización por pérdida de ventas la cual está en función de la tardanza respecto a la hora de inicio de la comercialización

representada por la variable *TAICH* que informa sobre la magnitud de la tardanza, así como sobre el intervalo en el que se da, ya que la penalización se compone de un elemento fijo dependiente del intervalo, y un componente variable dependiente de la magnitud.

La restricción (51) determina la tardanza o el anticipo del tiempo de llegada ($T_0 + dr$) respecto a la hora de inicio de descargues; la expresión (52) define el momento en que el vehículo termina de descargarse (*TD*) en función de la hora de inicio del descargue, el tiempo de servicio tiempo dependiente y la tardanza respecto a dicha hora de inicio del descargue, tal como se discute en la Figura 32; (53) calcula la tardanza o el anticipo respecto al inicio de la comercialización; en (54), (55) y (56) se asocia la magnitud de la tardanza frente a la comercialización (*TAIC*), a un intervalo *h*; (57) garantiza que a lo sumo exista un intervalo *h* asociado a *TAIC*; en (58) y (59) se define el intervalo *k* asociado al tiempo de llegada *TL*; (60) garantiza que se active a lo sumo un intervalo *k*; y la restricción (61) garantiza que el tiempo de descargue siempre sea menor a la hora en la que en el depósito se cierra.

Ahora, dado que el modelo requiere el parámetro de la duración del recorrido y que la simulación de la RdP permite obtener dicha duración para una determinada secuencia, entonces en este caso, para la etapa de evaluación, se usa la simulación de la red como un proceso previo tanto para cuando se entra a optimizar el tiempo de inicio con el modelo del OSTP, como para cuando el tiempo de inicio no se optimiza sino que viene dado y a partir del cual se requiere calcular la función de desempeño. Para el primer caso es claro que el valor de la duración de recorrido que sale de la simulación, entra como parámetro al modelo matemático, ahora resta por discutir el proceso para calcular la función de desempeño cuando no se optimiza.

La naturaleza de la función de desempeño hace que para el caso en el que no se optimiza, se requiere de un proceso lógico posterior a la simulación de la red, en el que se calculen los parámetros de los que depende la función de desempeño, a partir de la duración del recorrido. En la Figura 35 se muestra el pseudocódigo con el que se determina la función de desempeño, analizándola en tres componentes denotados como 1,2 y 3. El primer componente (1) corresponde al costo por consumo de gasolina durante el recorrido, cuyo cálculo es directo a partir del parámetro (*dr*) obtenido de la simulación de la RdP.

El segundo componente (2) es el costo por consumo de gasolina de la unidad de refrigeración, para lo cual se requiere determinar el tiempo de servicio (*ts*) correspondiente al tiempo de llegada ($T_0 + dr$), además del valor del *ANID*. El tercer componente (3) es el del costo por pérdida de ventas para lo que se necesita determinar el tiempo de descargue (*TD*) con el que se va a calcular la tardanza respecto a la hora de inicio de la comercialización *TAIC*, que posteriormente se usa para determinar tanto la pendiente como el costo fijo de la penalización.

Figura 35. Pseudocódigo para no optimizados

$$\text{CostoTotal} = \underbrace{(cgr \cdot cdg \cdot dr)}_{\textcircled{1}} + \underbrace{[cguf \cdot cdg \cdot (dr + ANID + \sum_k Y_k \cdot tstd_k)]}_{\textcircled{2}} + \underbrace{[\sum_h (TAICH_h \cdot pend_h + B_h \cdot ppv_h)]}_{\textcircled{3}}$$

① SimulaciónRdP $\rightarrow dr$

② $TL = T_0 + dr$

Para cada $k = 1$: cantidad de intervalos de tiempo de servicio

Si $TL \geq liits_k$ y $TL < lsits_k$, entonces :

$ts = tstd_k$

fin

fin

Si $hid > TL$, entonces :

$ANID = hid - TL$

$TAID = 0$

sino

$TAID = TL - hid$

$ANID = 0$

fin

③ $TD = hid + ts + TAID$

Si $liipv_1 > TD$, entonces :

$ANIC = liipv_1 - TD$

$TAIC = 0$

sino

$TAIC = TD - liipv_1$

$ANIC = 0$

fin

Si $TAIC = 0$, entonces :

$PEND = 0$

$PPV = 0$

sino

Para cada $h = 1$: cantidad de intervalos de pérdida de venta

Si $TAIC \geq (liipv_h - liipv_1)$ y $TAIC < (lsipv_h - liipv_1)$, entonces :

$PEND = pend_h$

$PPV = ppv_h$

fin

fin

fin

$$\text{CostoTotal} = (cgr \cdot cdg \cdot dr) + [cguf \cdot cdg \cdot (dr + ANID + ts)] + [PEND \cdot TAIC + PPV]$$

Elaboración propia.

Al implementar estos ajustes en el algoritmo propuesto en ésta tesis, se obtiene la herramienta metodológica de solución al problema, la cual se aplica al caso de estudio presentado en Osorio, Bravo-Bastidas, & Orejuela (2021) que se describe en lo que resta del capítulo, junto con los resultados obtenidos.

5.3 Resultados de aplicación

El marco general del problema abordado consiste en la recogida refrigerada de productos, posteriormente comercializados. Proceso que se caracteriza por factores particulares y relevantes como el costo de la unidad de refrigeración, así como los costos por pérdida de ventas, cuya consideración en la práctica para la programación de operaciones resulta importante para no caer en ineficiencias no deseadas. En el contexto colombiano, un ejemplo de ello es el caso de la comercialización de leche cruda (sin procesamiento) producida por los campesinos de Cumbal, municipio del departamento de Nariño, los cuales venden una parte de la producción a empresas de

procesamiento de leche, mientras que la producción restante puede ser comercializada en la galería municipal de Cumbal, para lo cual se requiere una recogida refrigerada del producto que garantice la cadena de frío requerida y que lleve el producto hasta el punto de venta en la galería.

Este reto logístico es compatible con el planteamiento del problema antes formulado, aplicable a perecederos, y en cuya aplicación práctica se tendría que el depósito corresponde a la galería donde se comercializa la leche, cuyos horarios de atención se representan con la ventana de tiempo discutida anteriormente. En este marco cobra sentido y relevancia la consideración de los costos asociados a la espera refrigerada, los tiempos de servicio tiempo dependientes y los costos por pérdida de ventas. Por consiguiente, dicha situación del municipio de Cumbal se propone como caso de estudio real para la aplicación de la metodología propuesta a la solución del problema.

5.3.1 Caso de estudio

En ésta sección se realiza una descripción del caso de estudio basada en el trabajo de Riascos & Taimal (2019):

El municipio de Cumbal se encuentra ubicado en el sur occidente del departamento de Nariño, cuenta con una economía agropecuaria, y su ubicación geográfica favorece principalmente la producción de leche. Esta actividad lechera se lleva a cabo fundamentalmente por productores organizados en asociaciones, las cuales venden aproximadamente el 75% de su producción a empresas procesadoras como Alpina, Colácteos y Alquería, y el 25% restante de la leche cruda requiere venderse en otros mercados (queseras o consumidor final) antes de que se cumpla el ciclo de vida del producto.

Las asociaciones adquieren enfriadoras con el objetivo de poder almacenar su producción diaria en las condiciones de frío necesarias. Según Riascos & Taimal (2019) hasta esa fecha existían 33 enfriadoras de leche; cada enfriadora es propiedad de los asociados pero recibe producto tanto de ellos como de proveedores (productores no asociados). Un operario es quien realiza la recepción de la leche y la dispone en el tanque frío para mantenerla en condiciones favorables. El número aproximado de socios por cada enfriadora son 43, generalmente con 10 proveedores. El proceso comienza en las fincas de los productores (socios y proveedores) los cuales realizan el ordeño y transportan el producto hasta la enfriadora de su asociación, y una vez el producto llega a la enfriadora se almacena en tanques de frío a una temperatura de 4°C.

Una de las principales problemáticas que enfrentan las enfriadoras es la brecha existente entre la oferta y la demanda, puesto que, como se mencionó anteriormente, aproximadamente el 25% de la leche almacenada en las enfriadoras no es comprada por los grandes clientes (plantas procesadoras), haciendo necesaria la búsqueda de otras alternativas para su venta, entre las cuales está la comercialización de la leche cruda en la galería municipal de Cumbal (Figura 36), lo cual requiere una gestión de recogidas eficiente que permita obtener un beneficio económico en esa búsqueda de expansión de mercados.

Figura 36. Galería municipal de Cumbal



Fuente: (Osorio et al., 2021).

Para el proceso de recogidas abordado en este caso de estudio, se van a considerar las coordenadas geográficas de 26 enfriadoras que Riascos & Taimal (2019) recolectaron en su investigación. La distribución geográfica en la zona rural del municipio de Cumbal se puede observar en la Figura 38 donde los puntos verdes corresponden a las enfriadoras y el punto rojo a la galería.

Para garantizar una adecuada calidad del producto y aumentar su vida útil, se requiere que tanto la recogida en las enfriadoras como el almacenamiento en la galería se haga de una manera refrigerada para conservar la cadena de frío que evita la rápida reproducción de microorganismos. Así que el proceso de recolección de la leche cruda se debe realizar en un vehículo refrigerado el cuál visita a cada una de las enfriadoras recogiendo el producto para posteriormente llevarlo a la galería para su comercialización. En este escenario, el depósito del tour es la galería y a ésta se le asocia una ventana de tiempo que empieza con la hora de inicio de los descargues de los vehículos (4:00 h o 14400 seg) y finaliza con la hora en la que se cierra el establecimiento (13:00 h o 46800 seg).

A partir de dicha ventana de tiempo surgen también otras particularidades del caso que son importantes en el ejercicio del ruteo en la recogida y que se plantean a continuación.

Costos de la espera refrigerada:

Al tener una recogida refrigerada y considerando la existencia de una hora específica en la que se inician los descargues de los vehículos en la galería, toma especial importancia la variable de adelanto o anticipo, definida como la diferencia entre la hora de inicio de descargue y la hora de llegada del camión, que se representa con la variable *ANID* discutida en la presentación de la metodología. La importancia de este tiempo radica en que si el camión llega antes de la hora de inicio de descargue (situación en la que la variable *ANID* toma valores), debe esperar a que llegue el momento de inicio de descargue y como la espera debe ser refrigerada, entonces existe un costo asociado al trabajo de la unidad de refrigeración del vehículo durante ese tiempo.

Cabe resaltar que el costo del uso de la unidad de refrigerado no ocurre únicamente en la mencionada espera, sino también en el recorrido mismo, y los parámetros necesarios para su cálculo son el consumo de gasolina de la refrigeración por unidad de tiempo, que es de 6.944×10^{-4} lt/seg y el costo de la gasolina que se asume de 2377.55 \$/lt, cuyos valores se definen según (AEOC, 2017).

Tiempo de servicio tiempo dependiente:

Adicionalmente, al tratarse de una galería en la que confluyen vehículos con productos para ser comercializados, ocurre que en las vecindades de la hora de inicio de los descargues, hay una alta congestión de vehículos para ser descargados, por lo que el tiempo de descargue de un vehículo que

recoge la leche depende de la hora en la que arribe al depósito. Para representar esta situación, se propone una función de tiempo de servicio como la mostrada en la curva roja de la Figura 33, en donde en cada intervalo se define una constante, es decir, hay un único tiempo de servicio asociado a cada intervalo, cuyos valores se detallan en la Tabla 5, sobre la cual se evidencia que el tiempo de servicio es mayor cuando la llegada del vehículo ocurre entre las 3:30 h y las 6:00 h, puesto que en esa franja se da la hora de inicio de descargue (4:00 h o 14400 seg), y alrededor de dicha hora es donde se espera mayor congestión.

Tabla 5. Función de tiempo de servicio.

Límite inferior del intervalo [seg]	Límite superior del intervalo [seg]	Tiempo de servicio [seg]
0 (0:00 h)	12600 (3:30 h)	3600
12600 (3:30 h)	21600 (6:00 h)	7200
21600 (6:00 h)	46800 (13:00 h)	2700

Fuente: (Osorio et al., 2021).

Penalización por pérdida de ventas:

Por otro lado, al tratarse de una comercialización del producto recogido que inicia una vez el camión es descargado, a medida que ese momento de descargue (variable TD presentada en metodología) ocurra después de la hora de inicio de la comercialización en la galería (5:00 h o 18000 seg), entra a jugar un costo asociado a la pérdida de ventas por falta de disponibilidad del producto, que se expresa como una función por partes, pero con la diferencia de que los segmentos asociados a cada intervalo no corresponden a una constante sino a una función lineal con pendiente positiva, tal como se observa en la Figura 34, ya que entre mayor sea la tardanza del descargue frente al inicio de la comercialización, mayor es la pérdida de ventas. Los datos tanto de la pendiente como de la constante del segmento de recta para cada intervalo se presentan en la Tabla 6.

Tabla 6. Función de pérdida de ventas

Límite inferior del intervalo [seg]	Límite superior del intervalo [seg]	Pendiente	Constante
18000 (5:00 h)	25200 (7:00 h)	0.5	80
25200 (7:00 h)	32400 (9:00 h)	0.8	110
32400 (9:00 h)	39600 (11:00 h)	1.1	150
39600 (11:00 h)	46800 (13:00 h)	1.3	180

Fuente: (Osorio et al., 2021).

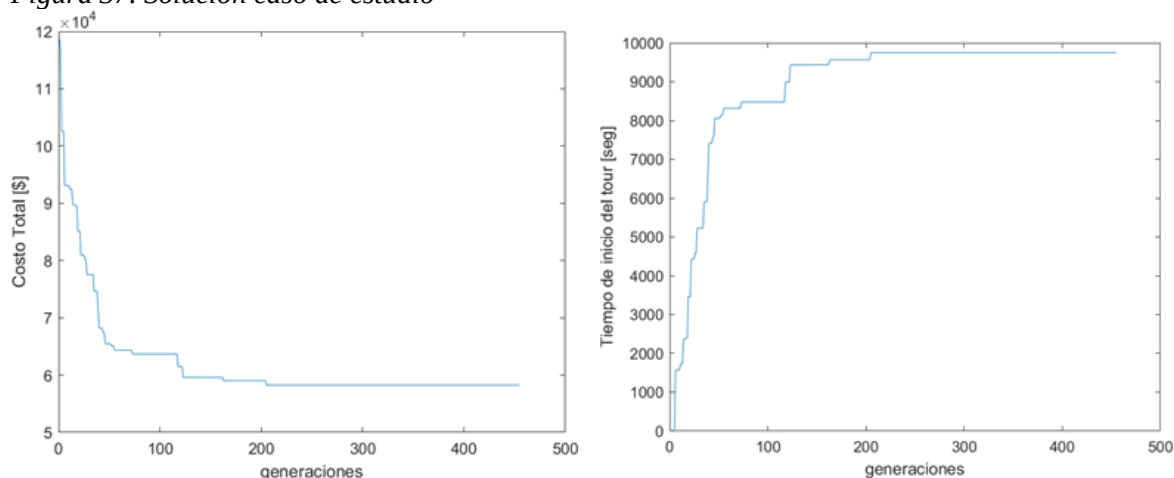
5.3.2 Análisis y discusiones

Partiendo de la descripción del caso de estudio y los datos presentados anteriormente, se parametriza el algoritmo propuesto como metodología de solución, cuyo objetivo es entonces, encontrar la ruta de las recogidas (secuencia de enfriadoras a visitar), así como también el tiempo de inicio del tour, tal que se minimicen los costos totales. Dicha función de desempeño se compone de, como se menciona anteriormente, costos por espera, costos por tiempo de servicio y penalización por pérdida de ventas, así como también del costo por consumo de gasolina en los desplazamientos entre nodos, para lo cual se requiere considerar el parámetro de consumo de gasolina por tiempo de recorrido que es de 0.0011 lt/seg (AEOC, 2017). Adicionalmente es necesario considerar, también, la velocidad promedio del vehículo para obtener el parámetro t_{ij} (tiempo de viaje del nodo i al j) a partir de la

matriz de distancias que se obtiene de las coordenadas de los nodos, que se encuentran en Riascos & Taimal (2019), y en este trabajo dicha velocidad promedio se asume de 45 Km/h.

El algoritmo propuesto fue programado en MATLAB® y se ejecutó con un tamaño de población de 1000 individuos, 500 generaciones, un porcentaje de elitismo del 2%, una probabilidad de mutación de 40% y una probabilidad de optimización del 60%, de lo cual se obtiene la siguiente solución: la secuencia de enfriadoras a visitar está dada por {0, 26, 20, 11, 5, 8, 23, 1, 17, 9, 22, 15, 10, 19, 6, 14, 2, 7, 12, 18, 16, 24, 25, 21, 13, 3, 4, 0}, donde 0 es la galería. El costo de la ruta fue de \$58127.7 con un tiempo de inicio de ruta de 9777.02 seg. En la Figura 37 se muestra el comportamiento, generación tras generación, tanto de la función de desempeño (gráfica de la izquierda) como del tiempo de inicio (gráfica de la derecha). Se puede observar que el algoritmo termina por convergencia en la generación 455.

Figura 37. Solución caso de estudio

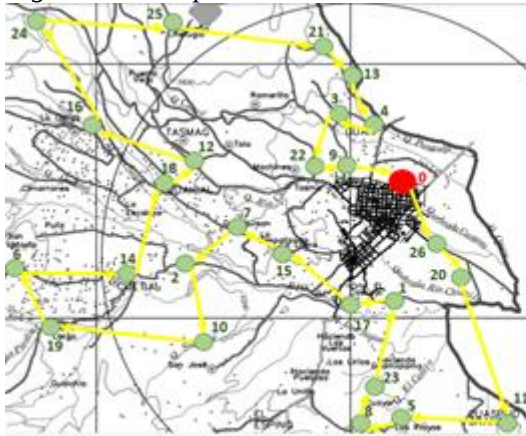


Elaboración propia.

Al tratarse de un algoritmo aproximado, las soluciones que se esperan son pseudo-óptimas. Y en este caso, para evaluar la calidad de la solución obtenida se recurre al artículo de Osorio, Bravo-Bastidas, & Orejuela (2021) en el que el mismo caso se resuelve de manera óptima a través de programación matemática, donde la solución está dada por la siguiente secuencia: {0, 26, 20, 11, 5, 8, 23, 1, 17, 15, 7, 2, 10, 19, 6, 14, 18, 12, 16, 24, 25, 21, 13, 4, 3, 22, 9, 0}, donde 0 es la galería. El costo óptimo de la ruta es de \$55093.5 con un tiempo óptimo de inicio de 10488.22 seg.

Comparando dicha solución óptima con la obtenida por el algoritmo propuesto, se evidencia la muy buena calidad de la solución, puesto que el gap de optimalidad respecto de la función de desempeño es de solo el 5.51%. Esta cercanía al óptimo se evidencia gráficamente en la Figura 38, donde se compara la ruta óptima con la ruta obtenida por el algoritmo, las cuales muestran una fuerte similitud.

Figura 38. Comparación soluciones



Solución Óptima



Solución pseudo-óptima

Fuente: basado parcialmente en (Osorio et al., 2021).

6. CONCLUSIONES Y TRABAJOS FUTUROS

A manera de conclusión se propone la siguiente síntesis del argumento general del trabajo, que se fue desarrollando en el marco de los objetivos específicos de la tesis, sobre un hilo conductor en el que se obtienen resultados parciales que actúan como premisas concatenadas desde las ideas básicas acerca de las ventajas de las RdP y los AG en problemas combinatorios, hasta las validaciones del algoritmo integrador cuyos buenos resultados lo sugieren como una metodología promisorio para aplicaciones futuras.

El trabajo inicia con la presentación general de las RdP y los AG como herramientas de modelamiento y de resolución de problemas, respectivamente. Particularmente en problemas combinatorios, son enfoques usados exitosamente, pero con limitantes o problemas cuando se aplican de manera independiente. Sin embargo, dichas limitantes se pueden resolver a través del uso conjunto y sinérgico entre las RdP y los AG, que es la idea esencial de esta tesis y cuyo desarrollo en la literatura científica apenas empezó a explorarse recientemente en el marco de problemas de scheduling.

La contribución de este trabajo es con un algoritmo de solución al TDTSP que integra redes de Petri y algoritmos genéticos. Para llegar a ello, primero se caracteriza el problema logrando una representación matemática, como se planeó en el primer objetivo específico, basada en el desarrollo que han tenido en la literatura las formulaciones del problema, las cuales iniciaron considerando la dependencia de los tiempos de viaje en términos de las posiciones en la secuencia, hasta llegar a formulaciones recientes que consideran la dependencia temporal en función de la hora o momento del día en el que el viaje se realiza, representando así, de una mejor manera fenómenos del tráfico urbano como lo son las denominadas horas pico.

El modelo planteado en esta tesis se basa en la más reciente consideración continua del tiempo, y al analizar dicho modelo se evidencia la mayor cantidad de variables y restricciones requeridas, respecto al TSP básico, además de la posibilidad de dividir el problema en dos sub-problemas secuenciales: el primero que es la definición de la secuencia de visitas, y el segundo, la determinación del tiempo de inicio óptimo para la secuencia (OSTP).

El OSTP se modela a partir de la formulación del TDTSP, y se utiliza dentro del algoritmo propuesto, el cual consiste en un algoritmo genético base cuya aplicación al problema se hace a partir de una representación en RdP del mismo, que se plantea como cumplimiento al segundo objetivo específico y que permite traducir el problema de secuenciación de nodos en el problema equivalente de secuenciación de disparos de transiciones, que es el que se propone resolver con el AG.

Para aplicar el AG se requiere, entonces, una estructura cromosómica acorde a la dinámica de disparos de transiciones de la RdP, la cual garantiza que cualquier individuo tanto generado aleatoriamente, como derivado a partir de los operadores genéticos, representa siempre una solución factible del problema. Dicha estructura cromosómica, así como su lógica de interpretación, se proponen en el capítulo 4. En este mismo capítulo se implementa dentro del algoritmo genético base, dicha estructura de interpretación de cromosomas, al igual que el OSTP, este último dentro de la etapa de evaluación del algoritmo, y con ello se obtiene un primer acercamiento a la propuesta metodológica de solución al problema, que se prueba en una instancia pequeña cuyos resultados permiten concluir que la ejecución exclusiva del OSTP como método de evaluación dentro del algoritmo, tiene un costo computacional elevado.

Con el objetivo de resolver ese problema del tiempo computacional del algoritmo, se propone el parámetro de “probabilidad de optimización”, para determinar en la etapa de evaluación del algoritmo cuáles individuos se optimizan y cuáles no. Lo anterior requiere agregar una lógica que permita calcular la función de desempeño en los casos en los que no se ejecuta el OSTP, respecto a lo cual el modelo en RdP toma especial relevancia, puesto que dada una secuencia de disparos y un tiempo de inicio, la simulación de la red permite el control estricto de la evolución temporal del problema en la secuenciación de las visitas, a partir de lo cual se puede calcular la función de desempeño.

Esta propuesta de la probabilidad de optimización no tiene precedentes en la literatura, lo cual exige validar el impacto dentro del algoritmo. Para ello se implementa la probabilidad de optimización y la lógica de simulación de la RdP dentro de la propuesta metodológica, y se realizan experimentaciones variando dicha probabilidad de optimización, encontrando que la calidad de las soluciones no se afecta cuando la probabilidad de optimización disminuye, mientras que se llegan a mejoras de hasta un 83% sobre el tiempo computacional, lo cual valida la pertinencia de la propuesta.

Con los ajustes anteriores, el algoritmo propuesto queda completamente diseñado, cumpliendo con ello el tercer objetivo específico, y permitiendo pasar a las pruebas con diversas instancias en donde se valida el buen comportamiento y desempeño. La calidad de las soluciones que brinda el algoritmo se comprueba en las instancias para las que se tiene el resultado óptimo como punto de comparación, y que el algoritmo obtiene sin problemas.

Las validaciones anteriores en instancias del problema, le dan cumplimiento al cuarto objetivo específico. Sin embargo, en el proceso investigativo se llegaron a unos resultados que desbordan el alcance de esta tesis pero que se deciden agregar por su aporte en la validación de la metodología de solución, respecto a la calidad de las soluciones, y a la flexibilidad de aplicación en distintos problemas de ruteo. En el capítulo 5 se presenta un caso real de ruteo con tiempo de servicio en depósito tiempo dependiente y con una función de penalización por pérdida de ventas también tiempo dependiente.

La propuesta metodológica de solución se basó en el algoritmo diseñado en la tesis, de lo cual se valida que la naturaleza tiempo dependiente del problema se aborda adecuadamente a través de la representación en RdP del problema, a la cual no es necesario hacerle ningún cambio, así como tampoco a la lógica de simulación de la red. Los únicos ajustes necesarios son en la etapa de evaluación del algoritmo, específicamente para los casos que no se optimizan es necesario agregar un proceso posterior a la simulación de la red, en el que se definen los parámetros requeridos para el cálculo de la nueva función de desempeño del caso de estudio. Por otro lado, respecto a los individuos para los que se optimiza el tiempo de inicio, es necesario replantear el modelo del OSTP, agregando las variables, parámetros y restricciones particulares del problema, así como la nueva función de desempeño.

Al implementar estos cambios para el caso de aplicación real, se procede a ejecutar el algoritmo obteniendo un resultado de muy buena calidad con un gap de tan solo el 5%. Todo lo anterior permite concluir la flexibilidad del algoritmo para ajustarse a problemas particulares, así como corroborar el buen desempeño mostrado en validaciones anteriores.

Por otro lado, el tiempo computacional siempre será un reto en estas metodologías aproximadas de solución, especialmente en instancias de gran tamaño, por lo que se recomienda explorar las opciones de computación paralela disponibles en MATLAB®, que se pueden aplicar en las etapas de interpretación de cromosomas, evaluación y mutación en donde la secuencialidad de un individuo a

otro no es necesaria, y por tanto se puede plantear la ejecución paralela de estas fases entre individuos, lo cual seguramente mejoraría aún más los tiempos de cómputo del algoritmo.

Adicionalmente, se propone complementar la estrategia de generación de tiempos de inicio no optimizados con un heurístico como lo puede ser la búsqueda local, cuyo costo computacional sea inferior a la optimización del modelo para el OSTP, con el objetivo de poder disminuir al mínimo posible la corrida de dicho modelo dentro del algoritmo, sin afectar la calidad de las soluciones, pero buscando disminuir el tiempo computacional.

Como trabajos futuros se identifican la aplicación de esta metodología a problemas más generales del ruteo tiempo dependiente, como los casos en los que se consideran tiempos de servicio tiempo dependiente, ventanas de tiempo y múltiple vehículos. Para estos casos el paradigma en redes de Petri es también aplicable y al ser problemas de mayor complejidad, el uso de las redes de Petri tendrá mayor impacto como complemento de los algoritmos genéticos.

BIBLIOGRAFÍA

- AEOC. (2017). Guía de cálculo de la huella de carbono del transporte de mercancías por carretera, 76. Retrieved from <http://www.comunidadism.es/wp-content/uploads/downloads/2019/03/Guía-Técnica-HC-mercancías.pdf>
- Ballou, R. (2004). *Logística. Administración de la cadena de suministro* (Quinta Edi). Pearson Educación.
- Barad, M. (2003). Tutorial and Survey Articles : An introduction to Petri Nets. *International Journal of General Systems*, 32(6), 565–582. <https://doi.org/10.1080/03081070310001623366>
- Bjorndal, M. H., Caprara, A., Cowling, P. I., Della Croce, F., Lourenço, H., Malucelli, F., ... Salazar, J. J. (1995). Some thoughts on combinatorial optimisation. *European Journal of Operational Research*, 83(2), 253–270. [https://doi.org/10.1016/0377-2217\(95\)00005-B](https://doi.org/10.1016/0377-2217(95)00005-B)
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, 35, 1–9. <https://doi.org/10.1007/s10479-005-3971-7>
- Brady, R. (1985). Optimization Strategies Gleaned from Biological Evolution. *Nature*, (315), 804–806.
- Caballero, J., & Mejía, G. (2006). Redes de Petri y algoritmos genéticos, una propuesta para la programación de sistemas de manufactura flexible. *Ingeniería Y Universidad*, 10(1), 4.
- Caballero, J. P., Mejía, G. E., & García, R. G. (2013). Scheduling of complex manufacturing systems with Petri nets and genetic algorithms : a case on plastic injection moulds. *International Journal of Advanced Manufacturing Technology*, 69(19), 2773–2786. <https://doi.org/10.1007/s00170-013-5175-7>
- Cacchiani, V., Contreras-Bolton, C., & Toth, P. (2020). Models and algorithms for the Traveling Salesman Problem with Time-dependent Service times. *European Journal of Operational Research*, 283(3), 825–843. <https://doi.org/10.1016/j.ejor.2019.11.046>
- Capkovic, F. (2012). Agent based approach to modelling ATM network. *IS'2012 - 2012 6th IEEE International Conference Intelligent Systems, Proceedings*, 102–107. <https://doi.org/10.1109/IS.2012.6335198>
- Castellanos, C. (2006). Consideraciones para el modelado de sistemas mediante Redes de Petri Aspects of modeling systems using Petri Nets. *Revisión Revista Ciencia E Ingeniería*, 27(2), 49–58. Retrieved from <https://www.redalyc.org/pdf/5075/507550778002.pdf>
- Chatterjee, S., Carrera, C., & Lynch, L. A. (1996). Genetic algorithms and traveling salesman problems. *European Journal of Operational Research*, 93(3), 490–510. [https://doi.org/10.1016/0377-2217\(95\)00077-1](https://doi.org/10.1016/0377-2217(95)00077-1)
- Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale Traveling Salesman Problem. *Journal of the Operational Research Society*, 2(4), 393–410.
- Duarte, A., Pantrigo, J., & Gallego, M. (2007). *Metaheurísticas*. (Dykinson, Ed.). Madrid.
- Essani, F. H., & Haider, S. (2018). An algorithm for mapping the asymmetric Multiple Traveling Salesman Problem onto Colored Petri Nets. *Algorithms*, 11(10).

<https://doi.org/10.3390/a11100143>

- Fox, K. . (1973). *Production scheduling on parallel lines with dependencies*. Johns Hopkins University.
- Fox, K. ., Gavish, B., & Graves, S. (1980). An n-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem. *Operations Research*, 28(4), 1018–1021. Retrieved from <http://dx.doi.org/10.1016/j.cirp.2016.06.001><http://dx.doi.org/10.1016/j.powtec.2016.12.055><https://doi.org/10.1016/j.ijfatigue.2019.02.006><https://doi.org/10.1016/j.matl et.2019.04.024><https://doi.org/10.1016/j.matlet.2019.127252><http://dx.doi.o>
- Giesen, R., Mahmassani, H. S., & Jaillet, P. (2005). Strategies for online inventory routing problem under real-time information. *Transportation Research Record*, (1923), 164–179. <https://doi.org/10.3141/1923-18>
- Giesen, R., Mahmassani, H. S., & Jaillet, P. (2010). *Logistics in real-time: Inventory-routing operations under stochastic demand. Lecture Notes in Economics and Mathematical Systems* (Vol. 619). https://doi.org/10.1007/978-3-540-92944-4_7
- Glover, F. (1986). Future Paths For Integer Programming And Links To Artificial Intelligence. *Computers and Operations Research*, 13(5), 533–549. [https://doi.org/http://dx.doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/http://dx.doi.org/10.1016/0305-0548(86)90048-1)
- Gouveia, L., & Voß, S. (1995). A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research*, 83(1), 69–82. [https://doi.org/10.1016/0377-2217\(93\)E0238-S](https://doi.org/10.1016/0377-2217(93)E0238-S)
- Grefenstette, J., Gopal, R., Rosmaita, B., & Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 160(168), 160–168.
- Gutin, G., & Punnen, A. (2003). *The traveling salesman problem and its variations*. Springer. <https://doi.org/10.16309/j.cnki.issn.1007-1776.2003.03.004>
- Halim, A. H., & Ismail, I. (2019). Combinatorial Optimization: Comparison of Heuristic Algorithms in Travelling Salesman Problem. *Archives of Computational Methods in Engineering*, 26(2), 367–380. <https://doi.org/10.1007/s11831-017-9247-y>
- Hashimoto, H., Yagiura, M., & Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, 5(2), 434–456. <https://doi.org/10.1016/j.disopt.2007.05.004>
- Holland, J. (1975). *Adaptation in Natural and Artificial System*. The University of Michigan Press.
- Huayna, A., Cortez, A., & Vega, H. (2009). Aplicación de las redes de Petri a la simulación discreta de sistemas. *Revista De Ingeniería De Sistemas E Informática*, 6, 35–44.
- Kamrani, A. K., & Gonzalez, R. (2003). A genetic algorithm-based solution methodology for modular design. *Journal of Intelligent Manufacturing*, 14(6), 599–616. <https://doi.org/10.1023/A:1027362822727>
- Kodama, A., & Nishi, T. (2017). Petri net representation and reachability analysis of 0–1 integer linear programming problems. *Information Sciences*, 400–401, 157–172. <https://doi.org/10.1016/j.ins.2017.03.014>

- Laporte, G. (2010). A concise guide to the Traveling Salesman Problem. *Journal of the Operational Research Society*, 61(1), 35–40. <https://doi.org/10.1057/jors.2009.76>
- Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2), 129–170. <https://doi.org/10.1023/A:1006529012972>
- Malandraki, C., & Daskin, M. (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science*, 26(3), 185–200.
- Mejia, G., Montoya, C., Cardona, J., & Castro, A. L. (2012). Petri nets and genetic algorithms for complex manufacturing systems scheduling. *International Journal of Production Research*, 50(3), 791–803. <https://doi.org/10.1080/00207543.2010.543177>
- Mejía, G., & Odrey, N. G. (2005). An approach using petri nets and improved heuristic search for manufacturing system scheduling. *Journal of Manufacturing Systems*, 24(2), 79–92. [https://doi.org/10.1016/S0278-6125\(05\)80009-3](https://doi.org/10.1016/S0278-6125(05)80009-3)
- Melgarejo, P. A., Laborie, P., & Solnon, C. (2015). A time-dependent no-overlap constraint: Application to urban delivery problems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9075, 1–17. https://doi.org/10.1007/978-3-319-18008-3_1
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM (JACM)*, 7(4), 326–329. <https://doi.org/10.1145/321043.321046>
- Moosavi, A., & Nikfarjam, A. (2019). A multi-path routing-inventory problem for a closed-loop supply chain considering the heterogeneous fleet of vehicles. *International Journal of Sustainable Engineering*, 12(3), 174–188. <https://doi.org/10.1080/19397038.2019.1566412>
- Mukhopadhyay, A., & Goswami, A. (2016). Stockout aversion in retailing supply chain using newsvendor models. *International Journal of Mathematics in Operational Research*, 8(2), 185–202. <https://doi.org/10.1504/IJMOR.2016.074854>
- Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77, 541–580.
- Neos Guide. (n.d.). Quadratic Assignment Problem. Retrieved from <https://neos-guide.org/content/quadratic-assignment-problem>
- Orejuela, J. P., & Hernández, G. (2019). Time Dependent School Bus Routing. *Revista Espacios*, 40(15), 24.
- Osorio, C. D., Bravo-Bastidas, J. J., & Orejuela, J. P. (2021). Ruteo con servicio tiempo dependiente y penalizaciones por espera y por pérdida de ventas. *Espacios*, 42(6), 117–135. <https://doi.org/10.48082/espacios-a21v42n06p09>
- Picard, J. C., & Queyranne, M. (1978). Time-Dependent Traveling Salesman Problem and Its Application To the Tardiness Problem in One-Machine Scheduling. *Operations Research*, 26(1), 86–110. <https://doi.org/10.1287/opre.26.1.86>
- Riascos, Z., & Taimal, O. (2019). *Diseño de un sistema de recolección y almacenamiento de leche para el municipio de Cumbal*. Universidad del Valle.

- Richard, P., Falcon Chang, M., Monmarche, N., & Proust, C. (1996). Visiting the traveling salesman problem with Petri nets and application in the glass industry. *IEEE Symposium on Emerging Technologies & Factory Automation, ETFA*, 1(1), 238–242. <https://doi.org/10.1109/etfa.1996.573298>
- Soto, M. (2007). *Métodos de optimización* (1st ed.). Madrid: Delta.
- Sun, C. (2020). A Study of Solving Traveling Salesman Problem with Genetic Algorithm. *ICITM 2020 - 2020 9th International Conference on Industrial Technology and Management*, 307–311. <https://doi.org/10.1109/ICITM48982.2020.9080397>
- Taş, D., Gendreau, M., Jabali, O., & Laporte, G. (2016). The traveling salesman problem with time-dependent service times. *European Journal of Operational Research*, 248(2), 372–383. <https://doi.org/10.1016/j.ejor.2015.07.048>
- Vahdati, G., Yaghoubi, M., Poostchi, M., & Naghibi S., M. B. (2009). A new approach to solve traveling salesman problem using genetic algorithm based on heuristic crossover and mutation operator. *International Conference of Soft Computing and Pattern Recognition*, 112–116. <https://doi.org/10.1109/SoCPaR.2009.33>
- Van Der Aalst, W. M. P. (1996). Petri net based scheduling. *OR Spectrum*, 18(4), 219–229. <https://doi.org/10.1007/BF01540160>
- Vélez Gallego, M., & Montoya, J. (2007). Metaheurísticos: Una alternativa para la solución de problemas combinatorios en administración de operaciones. *Revista EIA*, (8), 99–115. <https://doi.org/10.24050/reia.v4i8.188>