

ALGORITMO HÍBRIDO COMBINANDO UN SISTEMA
EVOLUTIVO COLONIA DE HORMIGAS CON BÚSQUEDA
TABÚ PARA LA SOLUCIÓN DEL PROBLEMA DE RUTEO
DE VEHÍCULOS CON VENTANAS DE TIEMPO DURAS

DEISY CAROLINA CANTILLO CALDERÓN

HENRY LAMOS DÍAZ Ph.D.
Director



GRUPO DE OPTIMIZACIÓN Y ORGANIZACIÓN DE
SISTEMAS PRODUCTIVOS, ADMINISTRATIVOS Y
LOGÍSTICOS OPALO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECAÑICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y
EMPRESARIALES
MAESTRÍA EN INGENIERÍA INDUSTRIAL
BUCARAMANGA
2014

ALGORITMO HÍBRIDO COMBINANDO UN SISTEMA
EVOLUTIVO COLONIA DE HORMIGAS CON BÚSQUEDA
TABÚ PARA LA SOLUCIÓN DEL PROBLEMA DE RUTEO
DE VEHÍCULOS CON VENTANAS DE TIEMPO DURAS

DEISY CAROLINA CANTILLO CALDERÓN

Proyecto de Investigación para optar el título de Master en
Ingeniería Industrial

Director
HENRY LAMOS DÍAZ Ph.D.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECAÑICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y
EMPRESARIALES
MAESTRÍA EN INGENIERÍA INDUSTRIAL
BUCARAMANGA
2014

*Dedicado a:
Dios
Mi madre
Humberto
Fernando Perez
Hermanos
Amigos*

TABLA DE CONTENIDO

INTRODUCCIÓN	11
1. VRPTW Y MÉTODOS DE SOLUCIÓN	13
2. ALGORITMO HÍBRIDO PARA EL VRPTW	21
2.1. VRPTW	21
2.2. FORMULACIÓN MATEMÁTICA	21
2.3. INSTANCIAS	24
3. DESARROLLO DEL ALGORITMO	27
3.1. Solución inicial: Heurística del vecino más cercano	28
3.2. Algoritmo colonia de hormigas	30
3.2.1. Colonia de hormigas VEI	32
3.2.2. Colonia de hormigas TIME	32
3.3. Heurística de inserción I1	34
3.4. Algoritmo búsqueda tabú	34
4. RESULTADOS ALGORITMO TACOLH	37
4.1. ANÁLISIS DE RESULTADOS OBTENIDOS	37
5. CONCLUSIONES	41
BIBLIOGRAFIA	43

LISTA DE FIGURAS

1.	Representación del VRPTW	23
2.	Formato de una instancia	25
3.	Framework	29
4.	Procedimiento colonia VEI [35]	33
5.	Procedimiento colonia TIME [35]	33
6.	Procedimiento vecindario	35
7.	Intensificación	36
8.	Comparación resultados por costos	38
9.	Comparación resultados por número de vehículos	39

LISTA DE TABLAS

- 1. Comparación de los métodos de solución 38
- 2. Resultados RC1 con menor número de vehículos 40

RESUMEN

TITULO:ALGORITMO HÍBRIDO COMBINANDO UN SISTEMA EVOLUTIVO COLONIA DE HORMIGAS CON BÚSQUEDA TABÚ PARA LA SOLUCIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS CON VENTANAS DE TIEMPO DURAS ¹

AUTOR:DEISY CAROLINA CANTILLO CALDERÓN ²

PALABRAS CLAVE: Optimización Combinatoria, Ruteo de vehículos con ventanas horarias, Colonia de Hormigas, Búsqueda Tabú, Heurística, Metaheurística

En el presente trabajo se aborda la problemática de la planificación de rutas de transporte para el reparto de mercancías donde se contempla la ventana de tiempo en el cual el cliente debe ser visitado, este tipo de problema es conocido en la literatura como el Problema de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW por sus siglas en inglés). El VRPTW aporta mayor realismo al problema de ruteo de vehículos. La complejidad de gran parte de los problemas de ruteo de vehículos hace que no se puedan resolver instancias en tiempo de cómputo razonable, es por ello que este tipo de problemas es considerado como NP-duros.

Para su solución, se han desarrollado técnicas aproximadas. Entre las herramientas de optimización, se encuentran los métodos metaheurísticos, que buscan soluciones en tiempo moderado. Se propone un algoritmo híbrido entre dos metaheurísticas: algoritmo colonia de hormigas y algoritmo búsqueda tabú, apoyándose en las heurísticas Vecino más cercano e Inserción I1. El algoritmo busca minimizar tanto vehículos como distancias, para ello trabaja con dos colonias de hormigas, VEI y TIME, cada una dedicada a un objetivo. Los resultados obtenidos son comparados con el conjunto de problemas tipo de Solomon (1987) y con otros encontrados en la literatura.

¹Trabajo de grado

²Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales. Director Henry Lamos Díaz

ABSTRACT

TITLE:HYBRID ALGORITHM COMBINING AN EVOLUTIONARY SYSTEM WITH ANT COLONY AND TABU SEARCH FOR THE SOLUTION OF THE VEHICLE ROUTING PROBLEM WITH HARD TIME WINDOWS ³

AUTHOR:DEISY CAROLINA CANTILLO CALDERÓN ⁴

KEYWORDS: Combinatorial Optimization, Vehicle routing problem with time windows, Ant Colony, Tabu search, Heuristic, Metaheuristic

In this research the problem of planning transportation routes for the delivery of goods is considered, where the time at which the customer must be visited is contemplated. This type of problem is known in the literature as The Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW brings more practicality to the Vehicle Routing Problem. However, due to the complexity of many of the vehicle routing problems, it makes that it cannot solve instances in a reasonable computation time; that is why this type of problem is considered as NP-hard.

For its solution, proximate techniques have been developed. Among the studied optimization tools, we have metaheuristic methods, which explore solutions in a moderate computational time. This present research proposes a hybrid algorithm between two metaheuristics: ant colony algorithm and tabu search, relying on the nearest neighbor heuristic and Insertion I1. The algorithm seeks to minimize both vehicles and distances, for it works with two ant colonies, VEI and TIME, each dedicated to a single objective. The results obtained are compared with the instances set of Solomon (1987) and other instances found in the literature.

³Research work

⁴Faculty of Physical-Mechanical Engineering. School of Industrial and Business Studies, Industrial Engineering. Advisor Henry Lamos Díaz

INTRODUCCIÓN

El transporte es una actividad fundamental y representa aproximadamente el 15% de los costos en las empresas [74]. Para el buen desarrollo de dicha actividad se debe tener presente todas las características de su entorno: modos de transporte de carga, geografía, tipos de usuarios entre otros. Por ejemplo, en Colombia resulta significativo la movilización por carretera, dado que por este medio se transporta aproximadamente del 80% de la carga del país ⁵. Por tal motivo los administradores han comprendido que un manejo eficiente de estas actividades contribuye al incremento de competitividad en las organizaciones.

Dado el impacto económico que tienen los problemas de transporte en el mundo empresarial, la comunidad de investigación de operaciones ha dirigido su interés hacia su solución mediante el uso de herramientas de optimización. El Problema de Ruteo de Vehículos (VRP) consiste en encontrar un conjunto de rutas desde un depósito para servir un número de clientes ubicados geográficamente con demandas conocidas a un mínimo costo. Los problemas de este tipo se pueden considerar problemas de optimización combinatoria, clase NP-hard, esto es, problemas que se cree no pueden ser resueltos de manera óptima mediante un algoritmo que pueda ser ejecutado en un tiempo polinomial.

Una de las variantes más populares del VRP se origina al incluir ventanas de tiempo para realizar recogida o entrega de mercancía a un grupo de clientes que se hallan distribuidos en una zona geográfica. Las ventanas de tiempo son periodos fijos durante los cuales se puede realizar la entrega o recogida de mercancía a los clientes, el problema es conocido como el VRPTW por sus siglas en inglés (Vehicle Routing Problem with Time Windows). Existen estudios que consideran las ventanas de tiempo de manera dura, esto es, ventanas de tiempo en las que no es posible realizar la entrega al cliente fuera de los periodos establecidos (VRPHTW), mientras que otros la consideran como ventanas de tiempo suaves, es decir, se permite la entrega fuera de estos periodos pero con una penalización (VRPSTW).

⁵Acosta de Valencia 2004

Debido a lo anterior, para una gestión efectiva del sistema de transporte es necesario el uso de programas de planificación, enfocados a la optimización de los procesos de distribución de la mercancía, donde el objetivo principal es encontrar rutas óptimas para minimizar tiempos y costos en el proceso de entrega y/o recogida y a su vez contribuyan de manera general a la reducción de los costos totales de la empresa.

En este documento se propone un nuevo algoritmo para la solución del problema de ruteo de vehículos con ventanas horarias, combina el algoritmo colonia de hormigas (ACO) según lo propuesto por [35] y el algoritmo búsqueda Tabú, con el propósito de mejorar las soluciones obtenidos por ACO y evitando los óptimos locales. Mediante la utilización de dos colonias de hormigas se busca el mejoramiento de las rutas en dos objetivos: la minimización del tiempo y de la flota vehicular.

El documento se encuentra ordenado de la siguiente forma: primero se encuentra la revisión bibliográfica realizada para la elaboración del trabajo de investigación, seguido por la formulación del VRPHTW, luego de la aplicación de las heurísticas ACO y Búsqueda Tabú al problema, después describe el híbrido propuesto en la investigación, siguiendo con el conjunto de resultados y el análisis de los mismos, finalmente, se encuentra la relevancia de la investigación a la comunidad académica.

1. VRPTW Y MÉTODOS DE SOLUCIÓN

El problema de ruteo de vehículos (VRP) fue trabajado por primera vez por [28] y desde entonces han sido extensivamente estudiados en el área de la investigación. Estos problemas tienen como objetivo diseñar un conjunto de rutas, una para cada vehículo de la flota, con mínimo costo, cumpliendo con una serie de restricciones dadas. De acuerdo a diferentes características que pueden presentar los clientes, los vehículos y el problema en general, surgen un conjunto de variaciones. En este proyecto de investigación se estudia la variante del VRP en la que cada cliente impone un tiempo en el cual debe ser visitado por el vehículo, conocido como Problema de Ruteo de Vehículos con Ventanas de Tiempo VRPTW. Estos problemas han sido investigados desde 1985, los primeros estudios sobre este tipo de problemas, y sus métodos de solución se pueden encontrar en [29], [44] y [68] proponen métodos de solución de manera exacta pero con tiempos de cálculos computacionales extensos.

En [67] se proponen métodos para la solución del VRPTW, los cuales han servido como base para diferentes investigaciones. El primer método propuesto en este estudio es una variación de la heurística de ahorro propuesta por [24]. También expone la heurística I1, que consiste en seleccionar un cliente semilla, aquellos que son los más distantes del depósito central o bien que proporcionan un instante de apertura más temprano para efectuar el servicio; y a partir del él se incorporan los siguientes hasta completar una ruta sin violar los requerimientos de capacidad del vehículo y de ventana de tiempo. En este trabajo se desarrolla un conjunto de problemas sobre las que valida las heurísticas propuestas. Dicho conjunto de Solomon ha servido como instancias benchmark para medir el rendimiento de los algoritmos propuestos por la comunidad de investigadores, con el propósito de normalizar los resultados.

En [59] presentan una versión paralela del método de inserción I1, al insertar el conjunto de rutas todas al mismo tiempo. Los autores utilizan la heurística de inserción de Solomon para determinar el número

inicial de rutas y el conjunto inicial de clientes semilla. En el 2001 [51] cambia el criterio de inserción secuencial propuesto por [67] para resolver un ejemplo práctico de una empresa de transporte de comida y un conjunto de problemas de la academia, el método estudiado en esta investigación busca la minimización del “impacto” sobre los clientes ya insertados, los que están a punto de ser insertados y los aún no insertados. Investigaciones realizadas por [34] y [59] propusieron mejoras a las primeras heurísticas desarrolladas en [67], enfocadas a la versión paralela del problema.

Las heurística de mejora parten de una solución inicial, generada de forma aleatoria o formada con base en las heurísticas de construcción como la del vecino más cercano o la heurística de ahorro; a partir de esta solución se realizan cambios en ella para mejorar las soluciones. [64] fue el primero en trabajar las heurísticas de mejora para el VRPTW a través de intercambios de arcos para conseguir soluciones k-óptimas. Posteriormente, en 1986 [4] realizan un estudio sobre los métodos de mejoramiento de rutas que pueden ser aplicados a los algoritmos de construcción. Las heurísticas utilizadas fueron 2-opt y 3-opt adaptadas al VRPTW.

En 1995, [60] comparan diferentes métodos de intercambios para la solución del VRPTW, como 2-opt, 3-opt y Or-opt, además, proponen una nueva heurística conocida como 2-opt*, las soluciones iniciales en el nuevo enfoque son generadas por el algoritmo I1 de Solomon. Por su parte, en [65] proponen la combinación de las heurísticas de construcción con las heurísticas de mejora, con base en el estudio se concluye que esta combinación es una opción más eficiente que la implementación de los algoritmos por separado, es decir, primero construir y luego mejorar.

En este mismo año, en [1] proponen un enfoque de construcción en paralelo que mejora múltiples rutas simultáneamente. El enfoque se basa en la concepción de negocio entre clientes y rutas vehiculares. En 1999, [50] proponen un algoritmo de dos fases que consiste en un método de inserción paralela para clústeres, así como un procedimiento de programación lineal de objetivos para generar las rutas. En la primera fase el algoritmo espera minimizar la distancia total recorrida en lugar del número de vehículos utilizados, mientras que el segundo criterio

busca la minimización del tiempo total de espera en los nodos. Los nodos semilla son seleccionados mediante la identificación de aquellos con limitaciones temporales. Los clientes que faltan se insertan de manera que el incremento de la distancia total de la ruta y el tiempo de espera sean mínimos. Al finalizar se mejora a través de los procedimientos de intercambio de 2-opt y Or-opt.

En [14] se describen heurísticas de búsqueda local, por ejemplo CROSS- y 2-opt*; utilizando un nuevo enfoque de tres fases para el VRPTW. En la primera fase, se generan las soluciones iniciales utilizando heurísticas de construcción de rutas a través de la combinación de diferentes parámetros. En la segunda fase se intenta reducir el número de rutas utilizando Eyección de Cadenas de nodos [40, 41] que también procura reordenar las rutas. En la tercera y última fase se utiliza el mecanismo de intercambio Or-opt para minimizar la distancia total recorrida. Uno de los mecanismos de construcción de rutas se fundamenta en los principios trabajados en [67] y [65], donde las rutas se generan de una en una de manera secuencial y después de que un número k de nodos hayan sido insertados en la ruta, se procede a la reordenación de los mismos a través del mecanismo Or-opt [16].

Por otra parte, existen los métodos aproximados llamados metaheurísticas que realizan una exploración más eficiente del espacio de soluciones en comparación con las heurísticas clásicas, en este tipo de algoritmo se utilizan procedimientos constructivos, algoritmos de búsqueda local, estrategias de combinación de soluciones y estructura de memorias. Los métodos metaheurísticos al ser más robustos crean la necesidad de usar recursos computacionales más eficientes para mejorar los tiempos de ejecución.

El algoritmo Búsqueda Tabú TS propuesto por Glover en 1986 consiste en realizar una búsqueda local aceptando soluciones que pueden llegar a deteriorar el valor de la función objetivo con la esperanza que en iteraciones posteriores se logre hallar mejores soluciones. La estrategia es utilizada para escapar de los óptimos locales. En 1994, [37] fue el primer trabajo en utilizar la metaheurística TS para solucionar el VRPTW, proponen una implementación paralela, intentan trasladar los clientes de aquellas rutas con menor número de nodos, hacia otras rutas. El método

genera una solución inicial aplicando la heurística I1 de Solomon, y a partir de esta solución aplican los procedimientos de mejora de 2-opt y Or-opt. Otra investigación en la que trabajan un método similar es [66], en la cual eliminan las rutas que tengan como máximo tres clientes a través de su inserción en otras rutas con mayor número de nodos.

Frecuentemente para encontrar la solución inicial los autores proponen utilizar un algoritmo de inserción, no obstante, trabajos como [23] aplican una versión heurística mejorada de la presentada en [65]; en [27] propone una versión cambiada del algoritmo de barrido [38] [2] y [66] utilizan el algoritmo de los ahorros de [24]. A su vez, para mejorar las soluciones iniciales se implementan heurísticas como 2-opt, Or-opt, 2-Opt*, k-intercambios, entre otras. Algunas investigaciones como las realizadas en [52] y [58] han demostrado que la heurística Or-Opt es eficiente para el VRPTW. En [63] presentan la llamada “memoria adaptativa”. Esta memoria es un grupo de rutas recogidas de las mejores soluciones encontradas durante la búsqueda. La estrategia proporciona nuevas soluciones de inicio para la búsqueda tabú mediante la selección y combinación de rutas tomadas de la memoria tabú.

En [23] modifican dinámicamente el tamaño de la lista de movimientos prohibidos para evitar los ciclos, aplican el método de λ - intercambios como proceso de búsqueda de vecindad. Durante este mismo año algunos autores proponen la utilización paralela en una red de computadores [3], en donde se obtienen mejoras en el tiempo de ejecución del algoritmo. Los autores de [66] también aplican una estrategia similar a la memoria adaptativa propuesta en [63] donde todas las rutas generadas en la búsqueda tabú se recogen en listas. Durante este mismo año, en [13] se describe un algoritmo de búsqueda tabú que permite soluciones factibles durante el proceso de búsqueda. Se halla la solución inicial mediante un algoritmo de inserción económica, en el método sólo los clientes cercanos a los ya incluidos en la ruta se seleccionan para ser insertados. La solución se puede mejorar mediante la inserción de clientes seleccionados al azar de otras rutas o en otro lugar con la misma ruta.

Los autores de [6] para el caso de estudio de recogida de residuos por cada vehículo, solucionan el problema a través de TS, agrupa a

los clientes por ventanas de tiempo compatibles, es decir, ventanas de tiempo cercanas y mejora los resultados mediante una heurística de vecindario. En el problema se considera que existe un conjunto significativo de restricciones relacionadas con el mundo real. En general se han tenido en cuenta las ventanas de tiempo asociadas con los clientes, las instalaciones de eliminación de residuos y el almacén. También tuvo en cuenta un período de descanso del conductor.

Otra metaheurística aplicada a la solución del VRPTW es Algoritmos Genéticos, en [11] se trabajó por primera vez en aplicar los algoritmos genéticos para la resolución del VRPTW. Para ello hibridaron un algoritmo genético con un algoritmo de tipo greedy. En esta aplicación, el algoritmo genético busca una buena programación de los clientes, mientras tanto el algoritmo de construcción genera las soluciones factibles para cada ruta.

El algoritmo en el que primero agrupan y luego rutean llamado GIDEON fue propuesto por [73], este asigna clientes a los vehículos para agrupar los clientes dentro de sectores con el algoritmo genético. Los clientes dentro de cada sector son enrutados usando la heurística de inserción económica, en el siguiente paso las rutas son mejoradas mediante la heurística λ -intercambios. Más tarde, [9] combinan un algoritmo genético con una heurística de construcción. La población inicial se construye con la heurística de inserción de un vecino más cercano en [67]. Los valores correspondientes a la aptitud de los individuos se basan en el número de rutas y la distancia total de la solución, omiten los problemas de codificación y representan una solución por medio de un conjunto de rutas factibles.

La mayor parte de los artículos aplican híbridos entre de los algoritmos genéticos con la aplicación de algún heurístico de construcción ([11]; [9]) de búsqueda local ([73], [72], [78] y [15]) así como otros metaheurísticos como la búsqueda tabú [47] y sistemas de colonias de hormigas [8]. Otra aplicación de híbrido se encuentran en [48], el algoritmo evolutivo se combina con una búsqueda tabú para minimizar la distancia total. [49] este trabajo es un híbrido compuesto por dos fases metaheurísticas para el problema de ruteo vehículo con ventanas de tiempo (VRPTW). El objetivo de la primera fase es la reducción del número de vehículos

por medio de un (μ, λ) - estrategia evolutiva, en la segunda fase para la reducción de la distancia recorrida se usa Búsqueda Tabú.

La Optimización Colonia de Hormigas es otra metaheurística utilizada para solucionar problemas combinatorios como el VRPTW, fue presentado por primera vez en [32] y propuesto en [35] para solucionar el VRPTW utilizan el enfoque fundamentado en la optimización por colonias de hormigas cooperativas [31]. La metaheurística optimiza en forma paralela dos objetivos, número de vehículos y distancia recorrida, mediante dos colonias de hormigas. La primera colonia denominada ACS-VEI, es la encargada de reducir el número de vehículos, mientras la segunda, ACS-TIME, se encarga de optimizar la distancia recorrida.

En [46] para solucionar el VRPTW se propone un algoritmo de colonias de hormigas en dos generaciones (padre e hijos). El objetivo de la generación de los niños primero es la construcción de la sub-tours, mientras que en la generación de segundo padre de la sub-tours se componen de soluciones factibles. Los resultados computacionales muestran que el algoritmo de colonias de hormigas de dos generaciones es eficaz en la solución de multi-objetivo VRPTW.

En el 2008, en [61] se propone como objetivo principal para el VRPTW la reducción del número de rutas o vehículos y estos autores opinan que la minimización del coste total del viaje es un objetivo secundario. En este trabajo se propone un algoritmo híbrido, llamado RPACS-VRPTW; entre ACO, Algoritmo aleatorio AR y búsqueda local de Pareto PLS. [7] también contempla las dos funciones objetivos, pero tiene como principal la de vehículo. Otra mejora para el ACO se muestra en [77] en este algoritmo la solución encontrada por cada hormiga puede ser mejorada mediante el algoritmo de barrido dinámico, después se aplica el ACO mejorado que actúa cuando el ACO se estanca.

La investigación realizada en [20] desarrolla un algoritmo híbrido entre el sistema de colonia de hormigas (IACS) propuesto por [19] y Recocido Simulado SA, llamado IACS-SA. El propósito fundamental de este enfoque es combinar la ventaja de IACS para encontrar buenas soluciones iniciales y la cualidad que posee el SA para salir de óptimos locales. De manera similar al híbrido propuesto por Chen C. y Ting C.

(2005) en [76] se propone construir un método que combina Optimización Colonia de Hormigas y Algoritmo Búsqueda Tabú, la dinámica de la combinación es igual ACO es bueno encontrando soluciones iniciales pero tiende a estancarse luego de cierto tiempo, para evitar este comportamiento se incorpora TS. Los métodos trabajan internamente con heurísticas tipo 2-opt.

Luego, [10] combina el ACO con búsqueda tabú. En primer lugar, el problema de fondo que se produce en la industria de la prensa, el problema de ruteo de vehículos con ventanas horarias y agrupaciones dependientes (VRPTWCD), implica la necesidad de cambios en los enfoques. En segundo lugar, la metaheurística ACO se puede utilizar para una computación eficiente de soluciones iniciales, pero necesita una búsqueda local (como sinónimo de búsqueda por vecindad) componente de optimización con el fin de lograr mejores soluciones. En tercer lugar, Tabu Search es una de las heurísticas más competitivas para los problemas de ruteo de vehículos con ventanas de tiempo. El resultado de la meta-heurísticos, llamado AntTabú, utiliza las ventajas de ACO (el cálculo eficiente de soluciones iniciales) y del TS (el de búsqueda local eficiente) y elimina los inconvenientes a través de una integración de ambos enfoques.

Otra combinación entre metaheurísticas es presentada en [70], el híbrido es entre Recocido Simulado y Búsqueda Tabú, capaz de explorar el espacio más grande de la búsqueda para superar el problema de los óptimos locales o ciclos de movimientos. Algunas de las aplicaciones más útiles de la VRPTW incluyen entregas bancarias [54], envíos por correo [56], recogida de residuos industriales [45], los servicios nacionales de franquicia de restaurantes [65], programación de rutas de autobuses escolares [12], la entrega de gases industriales [18] y JIT en fabricación [75].

Por otra parte, existe la optimización multi-objetivo que puede ser definida como el problema de encontrar un vector de variables de decisión que satisfacen restricciones y optimiza un vector de funciones cuyos elementos representan las funciones objetivo [25]. Los objetivos son independientes, por lo tanto la mejora o deterioro de uno no afecta a los otros. Por lo mencionado, no se encontrará una sola solución óptima para

el problema, debido a que cada solución óptima puede llegar a ser buena para un conjunto de objetivos pero no para otros.

A continuación se anuncian algunas aplicaciones, [69], [57] y [21] que considera el VRPTW como un problema de optimización bi-objetivo, reducir el número de vehículos y la distancia total del viaje; se utiliza un AG para la resolución del problema. Otros autores como [36] han trabajado el VRPTW multiobjetivo, plantea un algoritmo evolutivo mejorado multi-objetivo (MOEA) para resolver el problema de ruteo de vehículos con ventanas de tiempo (VRPTW). Minimiza simultáneamente el número de rutas, la distancia de viaje y tiempo de entrega. La mejora clave es la introducción de una medida de similitud entre las soluciones que se utiliza principalmente para seleccionar el segundo padre para el proceso de recombinación. Otra investigación basada en el problema multi-objetivo es [5], quienes minimizan simultáneamente la distancia recorrida y el desequilibrio de las rutas. El desequilibrio se evalúa desde dos perspectiva de las distancias recorridas, y la carga entregada por ellos. Un procedimiento se basa en recocido simulado.

Actualmente, se trabaja en nuevas variantes del VRPTW, las cuales contemplan características como vehículos heterogéneos [53] desarrollado mediante búsqueda tabú; combinación de tiempos de viaje estocásticos y ventanas horarias suaves resuelto por búsqueda tabú en [71] e incluso problemas donde se realiza entrega y recogida de mercancía simultáneamente, con flota heterogénea a clientes con ventanas de tiempo [55].

2. ALGORITMO HÍBRIDO PARA EL VRPTW

2.1. VRPTW

El VRPTW consiste en encontrar un conjunto de rutas con el menor costo de visitar n número de clientes geográficamente ubicados y con demandas asociadas d_i ; desde un depósito central, ubicado en el nodo 0. Las rutas diseñadas deben respetar las restricciones de ventanas de tiempo de los clientes, la ventana de tiempo del depósito, las restricciones de capacidad de los vehículos, los tiempos de servicio a los clientes y el número total de los vehículos.

Las aplicaciones de este tipo de problema en la vida cotidiana son múltiples, tales como: transporte de mercancía, transportes de residuos, rutas de buses escolares/laborales, etc. En Colombia empresas como Servientrega, Bavaria y Coca Cola ya usan software para optimización de la distribución; estos programas deben ser acoplados a las necesidades de cada una de ellas y buscando mejorar resultados en términos de costos y tiempo de planificación. Sin embargo, se aclara que aún en Colombia se tiene la desventaja de no tener cartografía de todo el territorio nacional para alimentar dichas herramientas ocasionando que el software no sea explotado al máximo. A continuación se presenta la formulación matemática que se utilizó, seguido de las características de las instancias trabajadas en el presente trabajo.

2.2. FORMULACIÓN MATEMÁTICA

Este trabajo utiliza la formulación matemática para el VRPTW con ventanas de tiempo duras presentada en [74]. El VRPTW se define por una red $G = (V, A)$ donde $V = \{0, 1, 2, 3, \dots, n\}$ es el conjunto de vértices que representan el depósito y los clientes. El depósito está representado por los nodos 0 o $n+1$. El conjunto $A = \{(i, j), i, j \in V\}$ es el conjunto de arcos de la red por los cuales viajarán los vehículos. Todos los vehículos inician y terminan su ruta en el depósito. Sea K el número de vehículos de igual capacidad. Cada cliente debe ser atendido sólo una vez, los clientes tienen asociada una demanda conocida, d_i ; $d_i \leq C$ para todo $i \in V$. El nodo depósito tiene asociada una demanda igual a cero, es

decir, $d_0 = 0$.

A cada cliente se le asigna un tiempo de servicio o ventana de tiempo $[a_i, b_i]$ dentro de la cual debe ser visitado, El valor inferior a_i es el tiempo más cercano para iniciar el servicio en el cliente i y b_i es el tiempo más lejano en que el vehículo debe iniciar el servicio al cliente i . Las ventanas de tiempo son duras, es decir, si el vehículo llega antes de iniciar la ventana, debe esperar, por lo contrario, si llega después de cerrada la ventana de tiempo no se le permitirá atender al cliente. El tiempo de viaje entre el nodo i y el nodo j es t_{ij} . La matriz de costos y de tiempos son simétricas. s_i es el tiempo que tarda el vehículo en atender al cliente. Se tiene una matriz de costos c_{ij} que representa el costo de ir del nodo i al nodo j . $\Delta^+(i)$ corresponde a los vertices que son directamente accesibles desde i y $\Delta^-(i)$ son los vertices a partir de los cuales i es directamente accesible. En la figura 1 se observa una solución al VRPTW con 10 clientes, la solución contiene 3 rutas así: $R_1 = [0, 1, 2, 3]$, $R_2 = [0, 4, 5]$ y $R_3 = [0, 6, 7, 8, 9, 10]$. N representa el número de clientes.

A continuación se presenta la formulación matemática para el VRPTW, en la que participan dos variables de decisión: w_{ik} es el instante de tiempo en que inicia el servicio en el cliente i por el vehículo k y

$$x_{ij} = \begin{cases} 1 & \text{si el arco } (i, j) \text{ hacen parte de la solución} \\ 0 & \text{en cualquier otro caso} \end{cases}$$

$$\sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad \text{Min} \quad (1)$$

sujeto a:

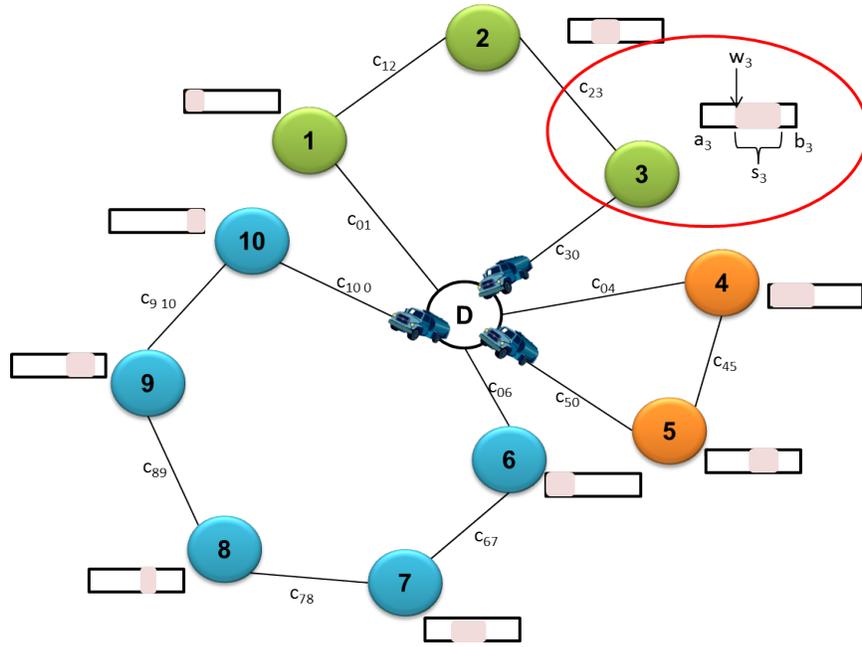


Figura 1: Representación del VRPTW

$$\sum_{k \in K} \sum_{j \in \Delta^+(i)} x_{ijk} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1 \quad \forall k \in K \quad (3)$$

$$\sum_{i \in \Delta^-(j)} x_{ijk} - \sum_{i \in \Delta^+(j)} x_{jik} = 0 \quad \forall k \in K, j \in N \quad (4)$$

$$\sum_{i \in \Delta^-(n-1)} x_{i,n+1,k} = 1 \quad \forall k \in K \quad (5)$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \leq 0 \quad \forall k \in K, (i, j) \in A \quad (6)$$

$$a_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq w_{ik} \leq b_i \sum_{j \in \Delta^+(i)} x_{ijk} \quad \forall k \in K, i \in N \quad (7)$$

$$E \leq w_{ik} \leq L \quad \forall k \in K, i \in \{0, n + 1\} \quad (8)$$

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq C \quad \forall k \in K \quad (9)$$

$$x_{ijk} \geq 0 \quad \forall k \in K, (i, j) \in A \quad (10)$$

$$x_{ijk} \in \{0, 1\} \quad \forall k \in K (i, j) \in A \quad (11)$$

La función objetivo que se presenta en (1), representa el costo total, el cual se puede interpretar como el tiempo de viaje o distancia recorrida total de todos los vehículos. Se desea encontrar la mínima distancia de recorrido total utilizando el menor número de vehículos. Las restricciones (2), indican que cada cliente será atendido por un sólo vehículo, las restricciones en (3) indican que para cada vehículo k , sólo un cliente j puede ser alcanzado desde el depósito, las restricciones en (4), muestran que el número de vehículos que llegan a un cliente es el mismo número de vehículos que sale, en (5) cada ruta tiene un sólo cliente que conecta al depósito.

En (6) garantiza la viabilidad de visitar al siguiente cliente en términos de tiempo. Las restricciones (7), indican que el instante de inicio del servicio, w_{ik} , debe iniciar dentro de la ventana de tiempo $[a_i, b_i]$. Restricciones en (8), imponen que el inicio del servicio en el cliente i debe empezar dentro de la ventana de tiempo $[E, L]$, este intervalo corresponde al tiempo de atención del depósito. Restricciones en (9), señala que la suma de las demandas de todos los clientes atendidos por el vehículo k no debe exceder la capacidad del vehículo. Las restricciones (10), son restricciones de no negatividad de las variables x . Restricciones (11), son restricciones que definen al problema como un modelo lineal entero binario.

2.3. INSTANCIAS

Las instancias o problemas tipos son conjuntos de datos que se usan para medir la calidad de los resultados de los algoritmos de

optimización propuestos; en [67] se crean un conjunto de 56 problemas que tienen diferentes características en cuanto a tamaños de las ventanas y demandas de los clientes. Cada uno de los 56 problemas se desarrollan para 100 clientes, un depósito central del cual deben salir y regresar al final de la ruta y una flota de 25 vehículos con capacidad limitada y homogéneas. En la Figura 2 se muestra la estructura de una instancia.

Nombre del problema						
Vehículos						
Número	Capacidad					
K	Q					
No. Cliente	Coord. x	Coord. y	Demanda	Tiempo más temprano	Tiempo más tardío	Tiempo de Servicio
0	x0	y0	q0	a0	b0	s0
1	x1	y1	q1	a1	b1	s1
2	x3	y3	q2	a2	b2	s2
.
.
.
100	x100	y100	q100	a100	b100	s100

Figura 2: Formato de una instancia

Los problemas están divididos en dos tipos según su distribución geográfica, los tipo C consisten en clientes ubicados en forma de clusters y el tipo R que tiene sus clientes ubicados aleatoriamente en el plano. A partir de la combinación de los dos anteriores se deriva el tipo RC, es decir, algunos clientes se encuentran ubicados aleatoriamente y otros clusterizados. Para cada tipo de problema se tiene dos conjuntos denotado por los número 1 y 2, hacen referencia a los horizontes de planificación pequeños y grandes (ventanas horarias). Cada categoría tiene una cantidad de instancias las cual se observan a continuación dentro de los paréntesis: C1 (09), C2 (08), R1 (012), R2 (011), RC1 (08) y RC2 (08).

Una de las características que hace que las instancias difieran una de la otra es el ancho de la ventana de tiempo, lo cual se define como la diferencia entre el fin el inicio de la misma, la ubicación geográfica de los clientes y la densidad de las ventanas, que no es más que el porcentaje de clientes que poseen ventanas horarias, por ejemplo, 25 %, 50 %, 75 % y 100 % de los clientes. Dado lo anterior, la notación de una instancia se construye de la siguiente manera: primero el tipo de problema C, R o RC; seguido del horizonte de planificación grupo 1 o 2 y el número del problema (01, 02,03 etc). Por ejemplo, R106 es el problema número seis de los clientes aleatoriamente georeferenciados y con horizonte de programación corto. Las instancias trabajadas en la investigación corresponden a las que poseen 100 clientes.

3. DESARROLLO DEL ALGORITMO

El problema de ruteo de vehículos es uno de los principales problemas en la optimización de operaciones logísticas que tiene como objetivo la reducción de costos. Para la solución de dichos problemas se han creado métodos aproximados, los cuales son una familia de algoritmos cuya meta es dar soluciones aproximadas a problemas tipo NP, sin tener que buscar en todo el espacio de soluciones. Existen dos tipos de heurísticas: constructivas y de mejora o búsqueda local; las de construcción parten del depósito y van visitando los clientes hasta encontrar una solución factible, son más ágiles pero proporcionan soluciones de baja calidad; los algoritmos de mejora o búsqueda local parten de una solución inicial creada normalmente con una heurística de construcción y la van reemplazando a medida que van encontrando mejores soluciones con sus estrategias.

El término metaheurística fue introducido por [39] “Las metaheurísticas son métodos aproximados diseñados para resolver problemas de optimización combinatoria, en los que los heurísticos clásicos no son efectivo”. Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos, combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos [42].

Las metaheurísticas, al igual que las heurísticas no aseguran una exploración sistemática de todo el espacio de soluciones, sin embargo, exploran aquellas regiones en las que se cree es posible encontrar buenas soluciones con base en ciertos criterios como puede ser el costo. Metaheurísticas bien diseñadas pueden evitar problemas de estancamiento en óptimos locales permitiendo encontrar mejores soluciones en comparación de las heurísticas clásicas. Su ventaja es la exploración más profunda en el espacio de búsqueda, pero, a costa de mayor tiempo de procesamiento. Algunas de las más conocidas en la literatura son: Algoritmo Colonia de Hormigas [30], Algoritmos Genéticos [43], Búsqueda Tabú [26] y Recocido Simulado [22]; más información sobre aplicaciones metaheurísticas se pueden encontrar en [17], [23], [22].

En el presente trabajo de investigación se desarrolla un algoritmo multiobjetivo: minimizar vehículos y distancia recorrida. Para lo anterior, se trabaja la combinación entre algoritmo colonia de hormigas propuesto por [35] y el algoritmo búsqueda tabú para la solución del VRPTW. Esta combinación de técnicas de solución se propone dado que en ocasiones el ACO deambula entre óptimos locales y cerca de la divergencia posee baja convergencia [33], [76] y por lo tanto es razonable introducir un método que prevenga el atascamiento del ACO en óptimos locales como lo es el algoritmo Búsqueda Tabú, el cual es una técnica eficiente para realizar búsquedas locales [10]. Para lograr optimizar vehículos y distancias se trabaja con dos colonias de hormigas artificiales, la primera es la encargada de minimizar los vehículos y la segunda la distancia recorrida. Se consideran cuatro partes importantes en el desarrollo del híbrido: colonia VEI, colonia TIME, construcción camino y búsqueda tabú; estas serán explicadas a lo largo del presente numeral. En la Figura 3 se observa el proceso del algoritmo propuesto.

3.1. Solución inicial: Heurística del vecino más cercano

El algoritmo comienza generando una solución inicial mediante la heurística del vecino más cercano, la cual se asume como mejor solución global, hasta tanto no sea mejorada por alguna de las colonias. Esta heurística está enfocada temporalmente, busca una solución basada en la cercanía que existe entre dos nodos adyacentes. Este enfoque posee una métrica que balancea entre la cercanía geográfica de los clientes y tiempo recorrido entre ellos. Es decir, un cliente que se encuentre geográficamente cercano no significa que sea el más factible en términos de tiempo, por lo anterior lo que busca esta dicha métrica es un balance entre estos dos parámetros.

Se tiene la siguiente ruta $(0, \dots, u_i, \dots, 0)$. Se busca insertar el cliente u_j después del cliente u_i para lo cual se calcula el costo de la inserción mediante la ecuación (12):

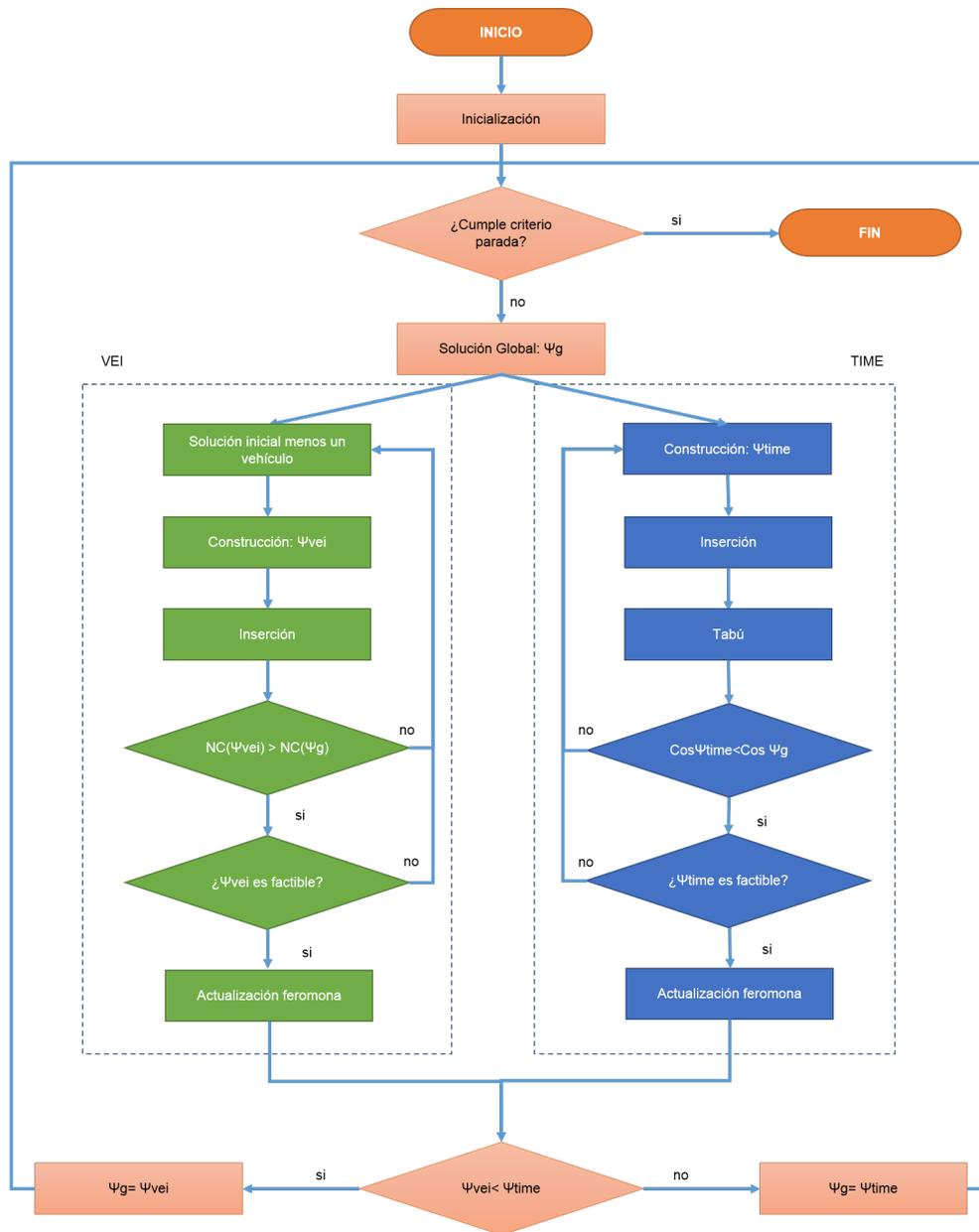


Figura 3: Framework

$$C_{ij} = \delta_1 d_{ij} + \delta_2 T_{ij} + \delta_3 V_{ij}, \quad \delta_1 + \delta_2 + \delta_3 = 0 \quad (12)$$

$$T_{ij} = W_j - (W_i + s_i) \quad (13)$$

$$V_{ij} = b_j - (2\theta_j + s_i + t_{ij}) \quad (14)$$

Donde δ_1, δ_2 y δ_3 son no negativos. d_{ij} corresponde a la distancia entre dos clientes, T_{ij} es la diferencia entre la hora de inicio del servicio en el cliente i y la hora de inicio del servicio en el cliente j y V_{ij} representa la urgencia de realizar la inserción. El valor de cada parámetro se asigna de acuerdo con el enfoque que se quiera dar al algoritmo. W_i representa el instante en el cual se inicia el servicio en el cliente i .

A partir del último nodo ubicado en la ruta se calcula los costos hacia cada uno de los clientes que aún no han sido visitados, ordenándolos en forma ascendente. Se selecciona el cliente respecto al costo, se debe tener en cuenta que sea factible su inserción y finalmente se agrega al final de la ruta. Si el primer cliente en la lista no es factible se toma el siguiente menor costo y así sucesivamente hasta encontrar un cliente factible. Este paso es iterativo hasta que ya no es posible insertar más clientes por alguna restricción, cuando sucede esto, el vehículo vuelve al depósito y se inicia un nuevo viaje. El proceso termina cuando todos los clientes son visitados.

3.2. Algoritmo colonia de hormigas

El algoritmo colonia de hormigas fue propuesto por [35] para la solución de VRPTW. Propone la creación de dos colonias de hormigas, con el fin alcanzar dos objetivos: minimizar número de vehículos y el tiempo de viaje. Esta minimización multiobjetivo es lograda mediante la creación de dos colonias de hormigas, una para cada objetivo. Para cada colonia se crean m hormigas artificiales. η_{ij} es definida como la información heurística, corresponde al inverso de la longitud del arco y es estática. Por otra parte, el nivel de feromona τ_{ij} es dinámica, cambia con el paso de cada hormiga, dicho nivel da una medida de deseo por seleccionar un arco e incorporarlo a la solución.

Cada hormiga k inicia en el depósito y construye una solución nodo a nodo, es decir, la hormiga escoge probabilísticamente un cliente entre un conjunto de clientes factibles N_i^k y lo va incorporando a la ruta hasta que todos los nodos han sido visitados. Para construir la ruta de la hormiga se usa la siguiente regla de probabilidad: con probabilidad q_0 se escoge

el nodo con más alto valor de $\tau_{ij}[\eta_{ij}]^\beta$, $j \in N_i^k$ y con $(1 - q_0)$ el nodo es elegido mediante la regla P_{ij} .

$$P(ij) = \begin{cases} \frac{\tau_{ij}[\eta_{ij}]^\beta}{\sum_{l \in N_i^k} \tau_{il}[\eta_{il}]^\beta} & \text{si } j \in N_i^k \\ 0 & \text{en caso contrario} \end{cases} \quad (15)$$

Donde β es el peso de la importancia del valor heurístico y q_0 determina la importancia de la explotación versus la exploración. Valores de q_0 pequeños da mayor probabilidad de utilizar la regla descrita en (15). Una vez la colonia de hormigas finaliza la construcción de su camino, la mejor solución encontrada se utiliza para actualizar los caminos de feromona. Después se inicia de nuevo el proceso con m hormigas hasta que la condición de parada se cumpla. Los criterios de parada más usados son: número fijo de soluciones generadas, tiempo de CPU transcurrido y mejora no encontrada luego de n iteraciones continuas.

En el algoritmo colonia de hormigas los rastros de feromona son actualizados localmente y globalmente. La primera es desarrollada a medida que la hormiga toma la decisión de ir de un nodo a otro y es usada para intensificar la búsqueda en el vecindario de la mejor solución (16), mientras que la actualización global es desarrollada al final de la construcción usando solo la mejor solución (17).

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (16)$$

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \frac{\rho}{J_\psi^{gb}} \quad (17)$$

Donde,

- $\rho \rightarrow (0 \leq \rho \leq 1)$
- $J_\psi^{gb} \rightarrow$ distancia de la solución global
- $\psi^{gb} \rightarrow$ solución global (ruta)
- $\tau_0 \rightarrow$ valor inicial feromona

$$\tau_0 = \frac{1}{nJ_\psi^h} \quad (18)$$

J_ψ^h es la distancia de la solución inicial y n es el número de clientes.

3.2.1. Colonia de hormigas VEI

La colonia de hormigas VEI es la encargada de encontrar la ruta que minimice el número de vehículos usados, maximizando el número de clientes visitados. La colonia toma como referencia el número de vehículos usados en la mejor solución global actual y obliga a las hormigas a encontrar una solución con un vehículo menos mediante la heurística del vecino más cercano, es decir, $v - 1$ vehículos. Dicha solución puede llegar a ser infactible, sin embargo, se asume como la mejor solución local. La solución es mejorada a medida que más clientes son visitados con menos vehículos.

Cada hormiga de la colonia construye una solución mediante la regla de probabilidad, puede que su solución al final no sea factible dada la restricción con los vehículos, sin embargo, luego de la construcción se aplica una heurística de inserción que ayudará a incluir a los clientes que no fue posible visitar y de esta manera aumentar el número de clientes visitados con menos vehículos. En la figura 4 de [35] se observa el algoritmo mediante el cual trabaja la colonia VEI.

Con el propósito de visitar cada vez más clientes la colonia de hormiga maneja un parámetro llamada IN, el cual es el encargado de almacenar las veces que el cliente j no ha sido incluido en la solución. Este parámetro ayuda en el procedimiento de construcción a dar más importancia a los clientes menos potenciales.

3.2.2. Colonia de hormigas TIME

En la colonia TIME las hormigas al igual que la colonia VEI inician construyendo la una solución ψ^h , luego de que cada hormiga de la colonia encuentra una solución, se busca mejorar el costo de dicha solución aplicando un procedimiento de búsqueda local como lo es la búsqueda tabú. Este procedimiento se desarrolla dado que el objetivos principal de la colonia TIME es la disminución de las distancias recorridas.

Después de ejecutar el algoritmo búsqueda tabú a la solución encontrada por las hormigas, se valida que de factible y es comparada con la mejor solución global. Si el resultado es mejor que la global, ésta

```

Procedimiento colonia VEI
1. Inicialización
Calculo de solución inicial local mediante vecino más cercano con un vehículo
menos  $\Psi^{vei}$ 
2. Ciclo
  Repita
    Para cada hormiga hacer
      /Construcción de solución/
      new_active_ant(k, tabú=falso, IN)
       $\forall$  cliente  $j \notin \Psi^k: IN_j \leftarrow IN_j + 1$ 
    fin para
      /actualización de la mejor solución si fue mejorada/
      si  $\exists k: \#clientes\ visitados(\Psi^k) > \#clientes\ visitados(\Psi^{vei})$  entonces
        enviar  $\Psi^k$  a principal y actualizar feromona
         $\tau_{ij} = (1 - \rho)\tau_{ij} + \rho / J_{\Psi^{vei}}^{vei} \quad \forall (i, j) \in \Psi^{vei}$ 
         $\tau_{ij} = (1 - \rho)\tau_{ij} + \rho / J_{\Psi}^{gb} \quad \forall (i, j) \in \Psi^{gb}$ 
      Fin del ciclo

```

Figura 4: Procedimiento colonia VEI [35]

es enviada a competir con las mejor solución encontrada en la colonia VEI, para que así alguna de las dos se convierta en el reemplazo de la global actual. Conocida una solución en TIME mejor que la global, se procede a la actualización de los rastros de feromona usando la ecuación 17.

```

Procedimiento Colonia TIME
1. Inicialización
2. Ciclo
  Repita
    Para cada hormiga hacer
      /Construcción de solución/
      new_active_ant(k, tabú=verdadero, 0)
    fin para
      /actualización de la mejor solución si fue mejorada/
      Si  $\exists k: \Psi^k$  es factible y  $J_{\Psi^k}^k < J_{\Psi}^{gb}$  entonces
        enviar  $\Psi^k$  a principal y actualizar feromona
         $\tau_{ij} = (1 - \rho)\tau_{ij} + \rho / J_{\Psi}^{gb} \quad \forall (i, j) \in \Psi^{gb}$ 
      Fin ciclo

```

Figura 5: Procedimiento colonia TIME [35]

3.3. Heurística de inserción I1

La heurística de inserción I1 fue creada en [67] inicia seleccionando un cliente semilla mediante un criterio determinado, por ejemplo, el más lejano al depósito, el cliente que posea la ventana de tiempo más próxima a cerrar o el cliente que minimice la suma del tiempo y distancia respecto al depósito. El criterio de inserción I1 considera el beneficio que se tiene al insertar en cliente u en la ruta que contiene los clientes i y j , en lugar de crear una ruta exclusiva para atenderlo. Se define como:

$$C_1(i, u, j) = \alpha_1 C_{11}(i, u, j) + \alpha_2 C_{12}(i, u, j), \quad \alpha_1 + \alpha_2 = 1 \quad (19)$$

$$C_{11} = d_{iu} + d_{uj} - \mu d_{ij}, \quad \mu \geq 0 \quad (20)$$

$$C_{12}(i, u, j) = W_{j_{new}} - W_j = PF(Push Forward) \quad (21)$$

$$C_2(i, u, j) = \lambda d_{0u} - C_1(i, u, j), \quad \lambda \geq 0 \quad (22)$$

Se calcula un C_1 para cada cliente u aún no visitado, lo cual determina la posición más económica en la que se puede insertar el cliente. Para medir el beneficio en términos de distancia en caso de insertar el cliente u entre i y j se calcula C_{11} . El Push forward es el tiempo adicional en el que se incurre al insertar el cliente u y se denomina como C_{12} . Para seleccionar el cliente a insertar se encuentra C_2 tomando el valor de C_1 y la distancia del cliente al depósito, teniendo como prioridad los clientes que serían más costoso al momento de tener que crearles rutas individuales. Por otra parte, si en el algoritmo se desea dar más peso a las variables temporales el valor de α_2 debe ser mayor y si lo más importante para el proyecto es optimizar las distancias recorridas la ponderación de α_1 debe ser la más alta que α_2 .

3.4. Algoritmo búsqueda tabú

La metaheurística búsqueda tabú fue introducida por Fred Glover, una descripción completa sobre la ella puede ser encontrada en [52]. Esta metaheurística busca escapar de óptimos locales usando algunas metodologías como lo son las heurísticas de intercambio.

Lista Tabú

Cada movimiento realizado en la búsqueda y que proporcione una mejor solución es guardado en una memoria a corto plazo llamada lista tabú. La lista tabú tiene como objetivo llevar un registro de los movimientos que pueden provocar que regresemos a los óptimos locales de las iteraciones anteriores, en ella se guardan los movimientos en forma de link producidos mediante la heurística 2-opt aplicada en el algoritmo. El tamaño de la lista tabú en el presente trabajo tiene una longitud de 5, la longitud de la lista debe ser como mínimo 1/10 de los movimientos posibles [62]. Vale la pena resaltar que se debe conservar la viabilidad de la solución después de cada movimiento, dado que es difícil volver una solución factible después de haber incumplido alguna restricción de tiempo.

Las estrategias de vecindad

Las estrategias de vecindad son las encargadas de generar nuevas soluciones para la TS, la estrategia aplicada en este algoritmo es la de intercambio, los pasos para el uso de la estrategia son:

- Luego de aplicar el algoritmo de inserción para los clientes no visitados en la colonia de hormiga y obtener una solución factible se realizan intercambios entre los diferentes clientes que comprendan la solución, excepto con el depósito. En el ejemplo a continuación el depósito es el nodo 0.

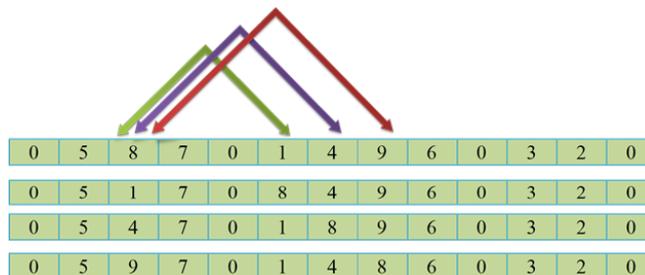


Figura 6: Procedimiento vecindario

- Al momento de tener todas las posibles soluciones derivadas de los intercambios, se debe validar que sean factibles y escoger aquella que presente el menor costo.
- La ruta de menor costo se asume como la nueva solución local. Este proceso es iterado hasta cumplir el criterio de parada.

Intensificación

La intensificación es la encargada de explorar con más detalle la mejora en la solución encontrada. Para ello, se divide la solución en subrutas y se aplica una búsqueda exhaustiva de mejora, en el caso de la presente investigación se aplica de nuevo la estrategia de intercambio; generando en estas nuevas soluciones factibles. Luego de encontrar mejores soluciones independiente, se unen de nuevo para generar una solución completa y factible.

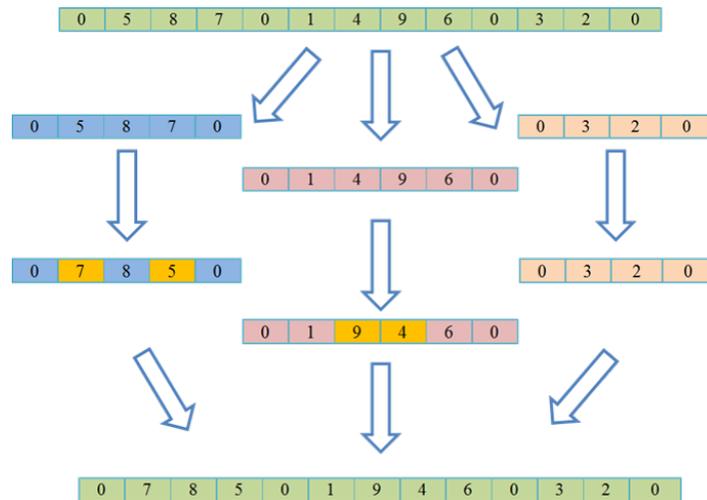


Figura 7: Intensificación

4. RESULTADOS ALGORITMO TACOLH

En el presente capítulo se muestran los resultados obtenidos en la implementación del algoritmo desarrollado en el capítulo tres. Se presenta la aplicación del algoritmo a las instancias propuestas en [67] y la comparación con otros resultados encontrados en la literatura. El algoritmo fue desarrollado en el software MATLAB® versión R2010a y ejecutado en computador portátil ASUS K43E con procesador intel® core(tm) i3- 2310M 2.10GHz y 3.00 GB de memoria RAM.

En este ítem se presentan los resultados proporcionados por algunos autores de los algoritmos heurísticos y metaheurísticos, para la solución de los problemas VRPTW. La solución se mide en términos del mínimo costo de la ruta y el número de vehículos; el costo está representado por la distancia recorrida por un vehículo, iniciando y terminando en el depósito. En la elaboración de la investigación una unidad de distancia es recorrida en una unidad de tiempo. La tabla 1 a continuación, muestra los resultados promedio obtenidos en los problemas tipo. En ella podemos observar la distancia (Dis) y número de vehículos (NV) promedios. Se trabaja con instancias de 100 clientes y vehículos con capacidad 200.

Para la heurística del vecino más cercano se usaron los parámetros $(\delta_1, \delta_2, \delta_3)$ y se realizaron corridas con: (0.4, 0.4, 0.2), (0, 1, 0), (0.5, 0.5, 0) y (0.3, 0.3, 0.4); sin embargo, aquellos que arrojaron mejores resultados fueron (0.4, 0.4, 0.2). Con los otros parámetros se observa la evolución de las soluciones pero con costos elevados e ineficientes en términos de uso de flota. En la heurística I1 los parámetros utilizados fueron $(\alpha_1, \alpha_2, \mu, \lambda)$ (1, 0, 1, 1). Finalmente, el algoritmo colonia de hormigas se desarrolla con $q_0 = 0,9$, $\beta = 1$, $\rho = 0,1$ y colonias de 5 hormigas.

4.1. ANÁLISIS DE RESULTADOS OBTENIDOS

En la tabla 1 se observa que en algunos casos el híbrido propuesto supera algunos de los resultados de otros autores. Las instancias tipo R1, R2, RC2, C2 y C1 aunque no presentan mejora respecto a los resultados de Solomon, se observa que respecto a Guillén (2002) es más

eficiente el propuesto en este trabajo en términos de número de vehículos.

AUTOR	R1		R2		RC1		RC2		C1		C2	
	Dis	NV	Dis	NV	Dis	NV	Dis	NV	Dis	NV	Dis	NV
Solomon (1987)	1437	13.6	1402	3.3	1597	13.5	1682	3.9	951	10	692	3.1
Potvin et al. (1993)	1509	13.3	1387	3.1	1724	13.4	1651	3.6	1343	10.7	797	3.4
Garcia et al. (1994)	1320	12.9	1229	3,1	1483	12,9	1551	3,9	877	10	602	3
Gambardella et al.(1999)	1217.7	12	967.8	2.73	1382.4	11.6	1129.2	3.25	828.4	10	589.9	3.0
Yu et al (2011)	1213.2	13.1	952.3	4.6	1415.6	12.7	1120.4	5.6	841.9	10	612.8	3.3
Guillén (2002)	1955	26	1239	8	2247	21	1573	11	1955	25	1485	14
Cantillo (2014)	1882	19.4	1677.3	3.8	2575	23	1994.6	4.5	2892	14	1431	3.6

Tabla 1: Comparación de los métodos de solución

Los costos se encuentran por arriba del promedio de los resultados encontrados en la literatura, sin embargo, en la figura 8 se observa la mejora en los problemas R1 Y C2 respecto a Gillén con menores desviaciones en los costos -3.71 y -3.65, respectivamente. En cuanto a número de vehículos usados, figura 9, en los problemas tipo 2 existe cercanía con los resultados de los otros autores.

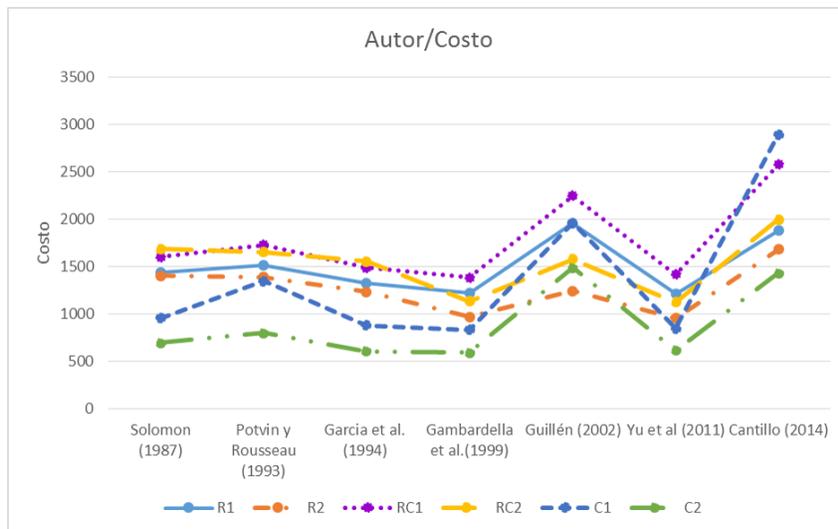


Figura 8: Comparación resultados por costos

El algoritmo TACOLH es evolutivo y por tanto una de las características que se quería evidenciar en él, era la evolución de las soluciones en los

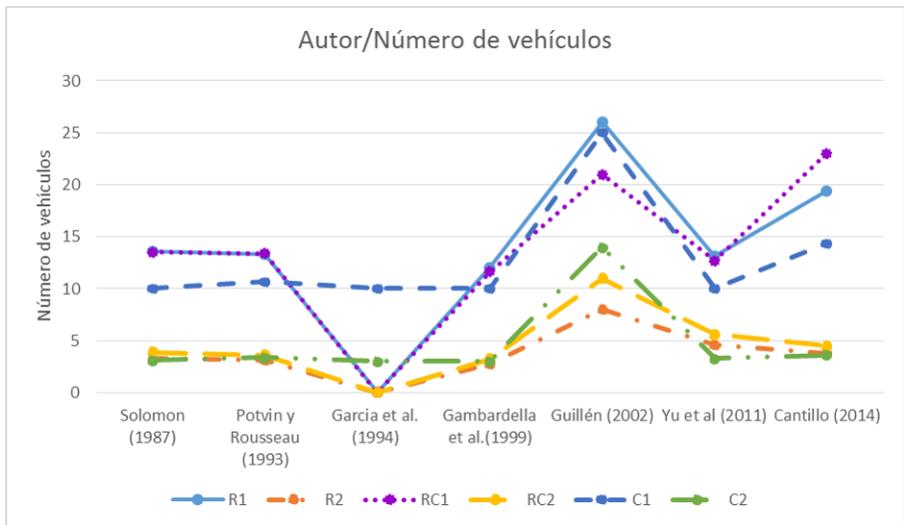


Figura 9: Comparación resultados por número de vehículos

diferentes problemas, desde la solución inicial hasta la final. De acuerdo a lo anterior, el algoritmo mostró resultados positivos en los problemas tipo 1, es decir, los problemas con horizonte de programación corto. Por otra parte, en los problemas con horizonte de programación largo, los tipo 2; no se logró ver evolución alguna, dado que los resultados obtenidos en la solución inicial fueron muy buenos y superarlos no fue posible para el híbrido.

La eficiencia del algoritmo depende de la solución inicial encontrada, pues esto permite tener un buen punto de partida para la evolución, sin embargo, dichas soluciones iniciales pueden ser limitantes a la hora de querer observar el progreso del algoritmo, dado que buenas soluciones iniciales encontradas serán difíciles de ser superadas y más aún cuando lo que se busca es una estrategia multiobjetivo, como era el caso de la presente investigación. Esta limitante se presentó en los problemas tipo 2.

El 52% de los ejercicios tipo 1 visitaron la totalidad de los clientes con menos vehículos de los usados en la solución inicial, no obstante, no fueron asumidos como mejor solución dado que su costo era superior al arrojado por la solución de la colonia TIME o por la encontrada en la solución inicial. En la tabla 2 se presentan los casos RC1 en los cuales

se puede evidenciar lo mencionado.

Además se observa que algunos casos, por ejemplo, RC104 Y RC107 logran visitar todos los clientes con menos vehículos en la colonia VEI pero no consiguen mejorar la solución en TIME mediante la aplicación de tabú (espacio vacío). Impidiendo alcanzar soluciones con menor distancia respecto a la solución inicial.

Instancia	DisIni	NVIni	DisFin	NVFin	DisVEI	NVVEI	DisTIME	NVTIME
RC101	2914.2	28	2640.9	25	3084.8	25	2640.9	25
RC103	3620.9	41	2861.6	24	3312	24	2861.6	24
RC104	1832	19	1832	19	2532.3	18	-	-
RC105	3677.5	37	3394	31	3579.8	31	3394	31
RC106	3030.2	28	2862.6	24	3044.6	24	2862.6	24
RC107	2424.9	23	2424.9	23	2897.9	22	-	-
RC108	2673.3	25	2424	21	3158.8	23	2424	21

Tabla 2: Resultados RC1 con menor número de vehículos

5. CONCLUSIONES

Luego de implementar el algoritmo TACOLH desarrollado sobre las instancias tipo, podemos concluir que la herramienta genera una nueva alternativa de solución, aportando una nueva metodología para la planificación de rutas. Una de las principales características del modelo es el haber encontrado buenas soluciones viables con pocas iteraciones.

Aunque las soluciones asumidas como mejores en el presente trabajo tiene prelación por las de menos costo, para las empresas es más importante el uso eficiente de la flota de vehículos que las distancias recorridas (siempre y cuando sean viables). Lo anterior se debe a lo representativo que son los costos de transporte para el área de distribución.

Aunque la optimización multiobjetivo trabajada en la presente investigación: distancia y número de vehículos muestra que es posible, es importante recalcar que no es fácil de alcanzar buenos resultados en los dos objetivos, dado que los objetivos son inversamente proporcionales .

Se niega la hipótesis de que los resultados obtenidos aplicando enfoque híbrido combinando un sistema evolutivo colonia de hormigas con Búsqueda tabú para la solución del Problema de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW) son iguales o mejores que los obtenidos hasta el momento en los problemas benchmarking de la literatura.

La generación de buenas soluciones iniciales no en todas las ocasiones permite evidenciar la evolución o mejoramientos de que se busca generar a partir de los algoritmos, dado que el espacio de posibles mejoras es reducido.

Se observa que la búsqueda del mejoramiento de las soluciones en términos de distancias y número de vehículos no es conveniente, ya que las soluciones conocidas en la literatura en este momento poseen excelente uso de la flota y forzar con esto a que la solución sea mejorada además con la distancia es poco posible. Sin embargo, si se pueden alcanzar mejoras en cuanto a distancias y tiempo de ejecución.

Por otra parte, se evidencia que dado el amplio estudio que se ha dado a este tipo de problemas a través de los diferentes métodos, ahora lo que se buscan son resultados por tipo de problema y no en conjunto.

Los resultados obtenidos muestran la combinación propuesta en el algoritmo muestra la evolución en las soluciones encontradas, sin embargo, se recomienda implementar otros métodos de búsqueda local e inserción que puedan mejorar aún más los resultados obtenidos.

Existe gran potencial de mejora en términos de tiempo computacional mediante el desarrollo de los algoritmos mediante procesamiento en paralelo CUDA. CUDA es una arquitectura de cálculo paralelo de NVIDIA que aprovecha la gran potencia de la GPU (unidad de procesamiento gráfico) para proporcionar un incremento extraordinario del rendimiento del sistema .

La aplicabilidad de los algoritmos desarrollados en las investigaciones es viable en casos de vida real, pero se aclara que en Colombia aún no se tiene cartografía adecuada para la implementación los mismos. Por otra parte, se observa un campo de desarrollo amplio en términos de contemplar las restricciones geográficas para la elaboración de rutas óptimas.

BIBLIOGRAFÍA

- [1] ANTES, J., AND DERIGS, U. A new parallel tour construction algorithm for the vehicle routing problem with time windows. Tech. rep., Universitet zu Koeln, 1995.
- [2] BACKER, B. D., FURNON, V., KILBY, P., PROSSER, P., AND SHAW, P. Local search in constraint programming: Application to the vehicle routing problem, 1997.
- [3] BADEAU, P., GENDREAU, M., GUERTIN, F., POTVIN, J., AND TAILLARD, E. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transactions Research* 5 (1997), 109–122.
- [4] BAKER, E., AND SCHAFFER, J. Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. *American Journal of Mathematical and Management Sciences* 6 (1986), 261–300.
- [5] BAÑOS, R., ORTEGA, J., GIL, C., FERNANDEZ, A., AND DE TORO, F. A Simulated Annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications* 40, 5 (Apr. 2013), 1696–1707.
- [6] BENJAMIN, A., AND BEASLEY, J. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research* 37, 12 (Dec. 2010), 2270–2280.
- [7] BERGER, J., AND BARKAOU, M. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO - Operations Research* 42, 3 (Aug. 2008), 415–431.
- [8] BERGER, J., BARKAOU, M., AND BRÄYSY, O. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Tech. rep., Defence Research Establsihment Valcartier, 2001.

- [9] BERGER, J., SALOIS, M., AND BEGIN, R. A hybrid genetic algorithm for the vehicle routing problem with time windows. Tech. rep., Vancouver, 1998.
- [10] BÖHNLEIN, D., GAHM, C., AND TUMA, A. A Hybrid Meta-Heuristic for the VRPTW with Cluster-Dependent Tour Starts in the Newspaper Industry. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on* (2009), IEEE, pp. 1–10.
- [11] BLANTON, J., AND WAINWRIGHT, R. Multiple vehicle routing with time and capacity constraints using genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms* (San Francisco, CA, USA, 1993), Morgan Kaufmann Publishers Inc., pp. 452–459.
- [12] BRACA, J., BRAMEL, J., AND SIMCHI-LEVI, D. A computerized approach to the new york cityschool bus routing problem. *IIE Transactions* 29 (1997), 693–702. 10.1023/A:1018526202990.
- [13] BRANDAO, J. Metaheuristic for the vehicle routing problem with time windows. 19–36.
- [14] BRÄYSY, O. A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. *INFORMS Journal on Computing* 15, 4 (Dec. 2003), 347–368.
- [15] BRÄYSY, O., BERGER, J., AND BARKAOUI, M. A new hybrid evolutionary algorithm for the vehicle routing problem with time windows. In *Presented at the Route 2000 workshop, Skoddsborg, Denmark* (2000).
- [16] BRÄYSY, O., AND GENDREAU, M. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science* 39, 1 (Feb. 2005), 104–118.
- [17] BRÄYSY, O., AND GENDREAU, M. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. *Transportation Science* 39, 1 (Feb. 2005), 119–139.

- [18] CAMPBELL, A., CLARKE, L., AND SAVELSBERGH, M. *Inventory routing in practice*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001, pp. 309–330.
- [19] CHEN, C., AND TING, C. An improved ant colony system algorithm for the vehicle routing problem. Tech. rep., Department of Industrial Engineering and Management, Yuan Ze University, 2004.
- [20] CHEN, C., AND TING, C. A hybrid ant colony system for vehicle routing problem with time windows. *Journal of the Eastern Asia Society for Transportation Studies* 6 (2005), 2822 – 2836.
- [21] CHIANG, T.-C., AND HSU, W.-H. A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. *Computers & Operations Research* 45 (May 2014), 25–37.
- [22] CHIANG, W., AND RUSSELL, R. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research* 63 (1996), 3–27. 10.1007/BF02601637.
- [23] CHIANG, W., AND RUSSELL, R. A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *Inform Journal on Computing* 9 (1997), 417–430.
- [24] CLARKE, G., AND WRIGHT, J. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12 (1964), 568–581.
- [25] COELLO, C. C. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In *Proceedings of the Congress on Evolutionary Computation* (1999), IEEE Press, pp. 3–13.
- [26] CORDEAU, J., AND LAPORTE, G. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the operation research society* 52 (2001), 928–936.
- [27] CORDEAU, J., LAPORTE, G., AND MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52 (2001), 928–936.

- [28] DANTZIG, G., AND RAMSER, J. The truck dispatching problem. *Management Science* 6 (1959), 80–91.
- [29] DESROCHERS, M., LENSTRA, J. K., SAVELSBERGH, M., AND SOUMIS, F. *Vehicle Routing: Methods and Studies*. Elsevier, 1988, ch. Vehicle Routing with Time Windows: Optimization and Approximation, pp. 65–84.
- [30] DORIGO, M. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.
- [31] DORIGO, M., CARO, G. D., AND GAMBARDELLA, L. Ant algorithms for discrete optimization. *Artificial Life* 5 (1999), 137–172.
- [32] DORIGO, M., MANIEZZO, V., AND COLORNI, A. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on system, man, and cybernetics-Part B* 26, 1 (1996), 29–41.
- [33] EL-SHERBENY, N. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science* 22, 3 (July 2010), 123–131.
- [34] FOISY, C., AND POTVIN, J. Implementing an insertion heuristic for vehicle routing on parallel hardware. *Computers and Operations Research* 20 (1993), 737–745.
- [35] GAMBARDELLA, L., TAILLARD, ., AND AGAZZI, G. Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In *New Ideas in Optimization* (1999), McGraw-Hill, pp. 63–76.
- [36] GARCIA, A., AND BULLINARIA, J. An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* 38, 1 (Jan. 2011), 287–300.
- [37] GARCIA, B., POTVIN, J., AND ROUSSEAU, J. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers and Operations Research* 21 (1994), 1025–1033.

- [38] GILLET, B., AND MILLER, L. R. A heuristic algorithm for the vehicle dispatch problem. *Operational Research* 22 (1974), 340–349.
- [39] GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13, 5 (1986), 533–549.
- [40] GLOVER, F. Multilevel tabu search and embedded search neighborhoods for the traveling salesman problem. Tech. rep., College of Business Administration, University of Colorado, 1991.
- [41] GLOVER, F. New ejection chain and alternating path methods for traveling salesman problems. *Computers & Operations Research* (1992).
- [42] GLOVER, F., AND KOCHENBERGER, G. *Handbook of Metaheuristics (International Series in Operations Research & Management Science)*. Springer, Jan. 2003.
- [43] GOLDBERG, D. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [44] GOLDEN, B., AND ASSAD, A., Eds. *Vehicle Routing: Methods and Studies*, vol. 16 of *Studies in Management Science and Systems*. North-Holland, Amsterdam, 1988.
- [45] GOLDEN, B., ASSAD, A., AND WASIL, E. *Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001, pp. 245–286.
- [46] GONG, W., LIU, X., ZHANG, J., AND FU, Z. Two-generation ant colony system for vehicle routing problem with time windows. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2007)* (2007).
- [47] HO, W., CHIN, J., AND LIM, A. A hybrid search algorithm for the vehicle routing problem with time windows. *International Journal on Artificial intelligence tools* 10 (2001), 431–449.

- [48] HOMBERGER, J., AND GEHRING, H. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR* 37 (1999), 297–318.
- [49] HOMBERGER, J., AND H.GEHRING. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European journal of operation research* 162, 1 (APR 1 2005), 220–238.
- [50] HONG, S., AND PARK, Y. A heuristic for bi-objective vehicle routing with windows constraints. *International Journal of Production Economics* 62 (1999), 249–258.
- [51] IOANNOU, G., KRITIKOS, M., AND PRASTACOS, G. A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of the Operational Research Society* 52 (2001), 523–537.
- [52] J. POTVIN, KERVAHUT, T., GARCIA, B., AND ROUSSEAU, J. The vehicle routing problem with time windows - part 1: Tabu search. *INFORMS* 8 (1996), 158–164.
- [53] JIANG, J., NG, K., POH, K., AND TEO, K. Vehicle routing problem with a heterogeneous fleet and time windows. *Expert Systems with Applications* 41, 8 (2014), 3748–3760.
- [54] LAMBERT, V., LAPORTE, G., AND LOUVEAUX, F. Designing collection routes through bank branches. *Computer's and Operations Research* 20 (1993), 783–791.
- [55] LOW, C., CHANG, C.-M., LI, R.-K., AND HUANG, C.-L. Coordination of production scheduling and delivery problems with heterogeneous fleet. *International Journal of Production Economics* 153, 0 (2014), 139 – 148.
- [56] MECHTI, R., POUJADE, S., ROUCAIROL, C., AND LEMARI, B. Global and local moves in tabu search: A real-life mail collecting application. Tech. rep., PriSM laboratory, University of Versailles, 2001.
- [57] OMBUKI, B., ROSS, B., AND HANSHAR, F. Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied intelligence* 24, 1 (FEB 2006), 17–30.

- [58] POTVIN, J., AND BENGIO, S. The vehicle routing problem with time windows - part 2: Genetic search. *INFORMS 8* (1996), 165–172.
- [59] POTVIN, J., AND ROUSSEAU, J. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eu 66* (1993), 331–340.
- [60] POTVIN, J., AND ROUSSEAU, J. An exchange heuristic for routeing problems with time windows. *Operational Research Society 46* (1995), 1433–1446.
- [61] QI, C., CUI, S., AND SUN, Y. Using Ant Colony System and Local Search Methods to Solve VRPTW. *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application* (Dec. 2008), 478–482.
- [62] RANGEL, D. F. G. C. E. *Formular las metaheurísticas búsqueda tabú y recocido simulado para la solución del CVRP (Capacitated vehicle routing problem)*. 2011.
- [63] ROCHAT, Y., AND TAILLARD, E. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics 1* (1995), 147–167.
- [64] RUSSELL, R. A. An effective heuristic for the m-tour traveling salesman problem with some side conditions. *Op 25* (1977), 517–524.
- [65] RUSSELL, R. A. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation S 29* (1995), 156–166.
- [66] SCHULZE, J., AND FAHLE., T. A parallel algorithm for the vehicle routing problem with time windows constraints. *Annals of Operations Research 86* (1999), 585–607.
- [67] SOLOMON, M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research 35, 2* (1987), 254–265.
- [68] SOLOMON, M., AND DESROSIERS, J. Time window constrained routing and scheduling problems. *Transportation science 22* (1988), 1–13.

- [69] TAN, K., CHEW, Y., AND LEE, L. A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Computational optimization and applications* 34, 1 (MAY 2006), 115–151.
- [70] TAN, K., LEE, L., ZHU, Q., AND OU, K. Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence un engineering* 15, 3 (JUL 2001), 281–295.
- [71] TAS, D., DELLAERT, N., VAN WOENSEL, T., AND DE KOK, T. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research* 40, 1 (Jan. 2013), 214–224.
- [72] THANGIAH, S. An adaptive clustering method using a geometric shape for vehicle routing problems with time windows. In *Proceedings of the 6th International Conference on Genetic Algorithms* (San Francisco, CA, USA, 1995), Morgan Kaufmann Publishers Inc., pp. 536–545.
- [73] THANGIAH, S. R. Vehicle routing with time windows using genetic algorithms, 1995.
- [74] TOTH, P., AND VIGO, D. *The vehicle routing problem*, vol. 9. Society for Industrial Mathematics, 2002.
- [75] VAIDYANATHAN, B., MATSON, J., MILLER, D., AND MATSON, J. A capacitated vehicle routing problem for just-in-time delivery. *IIE Transactions* 31 (1999), 1083–1092. 10.1023/A:1007631726356.
- [76] YU, B., YANG, Z. Z., AND YAO, B. Z. A hybrid algorithm for vehicle routing problem with time windows. *EXPERT SYSTEMS WITH APPLICATIONS* 38, 1 (JAN 2011), 435–441.
- [77] ZHEN, T., QIUWEN, Z., WENSHUAI, Z., AND ZHI, M. Hybrid Ant Colony Algorithm for the Vehicle Routing with Time Windows. *2008 ISECS International Colloquium on Computing, Communication, Control, and Management* (2008), 8–12.
- [78] ZHU, K. A new genetic algorithm for vrptw. In *Proceedings of the International Conference on Artificial Intelligence* (2000), p. 311264.