UN ALGORITMO ROLLOUT (RA) PARA LA SOLUCIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDA ESTOCÁSTICA, DESDE UNA PERSPECTIVA DE REOPTIMIZACIÓN

YESENIA KATHERINE NÚÑEZ NÚÑEZ

LESLIE PAOLA ROBINSON QUINTERO

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICO-MECANICAS

ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES

BUCARAMANGA

2014

UN ALGORITMO ROLLOUT (RA) PARA LA SOLUCIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDA ESTOCÁSTICA, DESDE UNA PERSPECTIVA DE REOPTIMIZACIÓN

YESENIA KATHERINE NÚÑEZ NÚÑEZ

LESLIE PAOLA ROBINSON QUINTERO

Plan de proyecto de grado para optar al título de

Ingeniero Industrial

Director

HENRY LAMOS DIAZ

Matemático, Ph.D.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICO-MECANICAS

ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES

BUCARAMANGA

2014

DEDICATORIA

A la providencia divina por guiar mis pasos durante mi existir

A mis padres por brindarme su apoyo incondicional a pesar de las adversidades

A mi hermana por creer en mí en todo momento, por ser un ejemplo y estar siempre a mi lado

A mi sobrino José David por existir, por ser mi inspiración, por traer felicidad a mi vida y a la familia

A Juan Carlos Acevedo por ser una motivación en mi vida y acompañarme en el cumplimiento de tan esperado logro

A mi compañera Leslie Robinson, porque jamás nos dimos por vencidas

YESENIA KATHERINE NÚÑEZ NÚÑEZ

DEDICATORIA

A Dios por escuchar mis oraciones y permitir cumplir una etapa más en mi vida.

A mi mamá por estar siempre conmigo, siendo la motivación más grande que tengo para salir adelante.

A mi tía y a mi abuela por su apoyo y cariño incondicional en todo momento.

A María Camila por su amor y confianza constante.

A mis amigos por acompañarme en este proceso.

A Fabian Alean, por su amor, apoyo y comprensión.

A mi compañera Yesenia Núñez por compartir esta experiencia conmigo.

LESLIE PAOLA ROBINSON QUINTERO

AGRADECIMIENTOS

A Dios por permitir realizar este proyecto y que hoy sea una realidad.

Al Doctor Henry Lamos Díaz por confiar en nosotras y guiarnos en el desarrollo de nuestro proyecto.

A la Ingeniera Silvia Galván Núñez por las guías brindadas, útiles para el desarrollo de esta investigación.

A la Ingeniera Clara Novoa por la asesoría brindada en el presente proyecto de investigación.

Al Ingeniero Juan Carlos Acevedo Pinzón por su tiempo, paciencia y su colaboración a lo largo de todo el proyecto.

Al Ingeniero Leonardo Jaimes por su dedicación y tiempo.

A la Universidad Industrial de Santander, Escuela de Estudios Industriales y Empresariales y grupo de investigación OPALO por la formación académica.

Yesenia Katherine Núñez Núñez

Leslie Paola Robinson Quintero

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

TÍTULO: "UN ALGORITMO ROLLOUT PARA LA SOLUCIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDA ESTOCASTICA, DESDE UNA PERSPECTIVA DE REOPTIMIZACIÓN"[1]

AUTORES:
NÚÑEZ NÚÑEZ, Yesenia Katherine
ROBINSON QUINTERO, Leslie Paola[2]

PALABRAS CLAVES:
VRP, VRPSD, Programación Dinámica, Algoritmo de Despliegue, Secuencia de Base.

DESCRIPCIÓN

En el presente trabajo de investigación se presenta el Algoritmo Rollout de un solo paso (ORA) como una solución al Problema de Ruteo de Vehículo con Demanda Estocástica (VRPSD) con una perspectiva de reoptimización para el caso de único vehículo, el cual se centra en hallar una política mejorada que minimice el costo de la ruta resultante. Esto se obtiene por mejoras realizadas secuencialmente y de manera cíclica, teniendo como base una solución a priori hallada a través de la heurística Nearest Neighbor (NN).

Se realizó un diseño de experimentos con el objetivo de analizar los efectos de la variación de los parámetros del algoritmo Rollout sobre la variable respuesta. Utilizando la propuesta de investigación realizada por Silvia Galván se ha construido un banco de pruebas, adaptando los factores y niveles al presente problema para validar el Algoritmo Rollout. Se validaron las soluciones obtenidas por el software y se medió la calidad de los resultados, los cuales fueron comparados con el método evolutivo de colonia de hormigas (EACO).

A partir del resultado del diseño de experimentos se concluye que, el análisis del algoritmo se dividió para 4, 8 y 12 clientes. En los efectos estimados, para 4 y 8 clientes la ubicación de los clientes y la demanda promedio respectivamente, son factores significativos sobre la variable de respuesta. Para obtener mejores resultados sobre la función objetivo la ubicación de los clientes y la demanda promedio deben tomar el nivel 1 mientras que la desviación debe tomar el nivel 2. Los resultados obtenidos al comparar el Rollout con el método EACO muestran que el 37,5% de las instancias, para el algoritmo Rollout, arroja mejores resultados en el grupo de 8 y 12 clientes, con una diferencia porcentual mayor al 1%.

[1] Proyecto de Grado
[2] Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales. Director: Ph. D. Henry Lamos Díaz

ABSTRACT

TITLE: "AN ROLLOUT ALGORITHM FOR THE SOLUTION OF THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMAND, FROM PERSPECTIVE OF REOPTIMIZATION*

AUTHORS:
NÚÑEZ NÚÑEZ, Yesenia Katherine**
ROBINSON QUINTERO, Leslie Paola

KEYWORDS:
VRP, VRPSD, Dynamic Programming, Algorithm Rollout, Base Sequence

DESCRIPTION

The current investigation presents the one-step rollout algorithm as a solution to the vehicle routing problem with estochastic demand (VRPSD) with a reoptimization perspective for an only-vehicle case, which is centered in finding an improved policy to minimize the cost of the resulting route. This is obtained by the upgrades made sequentially and cyclically, having as a foundation an a-priori solution found through the nearest neighbor heuristic.

Experiments were design to analyze the effect of variations of the parameters from the rollout algorithm on the variable answer. Using the investigation proposal made by Silvia Galvan a test bank has been built, adapting the factors and levels to the current problem to ratify the Rollout algorithm. The solutions obtained were validated by the software and the quality of the results was measured, which were compared using *Evolutionary Method based on Ant Colony Optimization,* EACO.

Based on the result of the design of experiments it is concluded that the analysis of the algorithm is divided to 4, 8 and 12 clients. In the estimated effects, 4 and 8 clients the location of customers and the average demand respectively, are significant factors on the response variable. For best results on the objective function the location of customers and average demand should take level 1 while the deviation should take Level 2. The results obtained by comparing the Rollout with EACO method show that 37.5% instances, for the Rollout algorithm yields better results in the group of 8 to 12 guests, with a percent difference greater than 1%.

.

* Graduation Project
** Faculty of Engineering physicomechanical. School of Industrial and Business Studies. Director: Ph D. Henry Diaz Lamos.

INTRODUCCIÓN

El problema de ruteo de vehículos (*Vehicle Routing Problem*, VRP) consiste en el diseño de rutas eficientes utilizando técnicas y algoritmos propios de la investigación de operaciones. El diseño de rutas vehiculares es hoy en día para pequeñas, medianas y grandes empresas, una forma de ser más competitivos al poder ofrecer un óptimo servicio y así reducir costos de transporte y satisfacer a los clientes en los tiempos de entrega. Por consiguiente, es un eslabón importante en la cadena de suministro.

El VRP consiste en hallar un conjunto de rutas que permitan satisfacer las demandas de clientes localizados en una zona geográfica y cada uno de los vehículos presenta una capacidad limitada Q. El objetivo es minimizar costos, comenzando y terminando el recorrido en el depósito.

Las restricciones dadas por el entorno, dan lugar a las diferentes variantes del VRP en la cual se encuentra el problema de ruteo de vehículo con demandas estocásticas (*Vehicle Routing Problem with Stochastic Demands*, VRPSD) en donde, la demanda de cada cliente se considera como una variable estocástica con distribución de probabilidad conocida. El VRPSD pertenece a la categoría de problemas combinatorios NP-Hard, por lo que no se garantiza que se encontrará una solución óptima en un tiempo computacionalmente adecuado; así que en esta clase de problemas es necesario usar algoritmos aproximados desde una perspectiva de reoptimización.

El objetivo del presente trabajo de investigación es aplicar un algoritmo Rollout usando la programación dinámica (PD) para la solución del VRPSD. Un algoritmo Rollout es un método aproximado que resuelve grandes problemas de PD únicos en tiempo real. La importancia de esto reside en el hecho de estudiar el problema estocástico de ruteo de vehículo mediante otros enfoques diferentes a las heurísticas y determinar las ventajas y desventajas de su utilización.

1. PLANTEAMIENTO DEL PROBLEMA

El problema de ruteo de vehículos (*Vehicle Routing Problem*, VRP) consiste en hallar un conjunto de rutas que atienda o recoja mercancía a clientes localizados en una zona geográfica, cada cliente tiene una demanda que debe satisfacerse. Se considera que cada uno de los vehículos tiene una capacidad limitada Q. Las rutas deben ser aquellas que minimicen los costos, comenzando y terminando el recorrido en cierta localización llamada depósito. La demanda de cada cliente se puede considerar como una variable estocástica con distribución de probabilidad conocida; en este caso el problema de ruteo se llama problema de ruteo estocástico (SVRP, por sus siglas en ingles).

En el presente proyecto se estudiará el problema de ruteo de vehículos con demandas estocásticas (*Vehicle Routing Problem with Stochastic Demands,* VRPSD), cada cliente presenta una demanda que sigue una distribución de probabilidad discreta y ésta solo se revela cuando el vehículo llega a la locación del cliente[3]. Puede presentarse que a lo largo de la ruta establecida el vehículo supere su capacidad o quede sin existencias, por lo tanto se ve obligado a acudir al depósito para dejar o reabastecerse y volver a su ruta para continuar con el recorrido; ante esta situación se aumentan los costos de enrutamiento y la distancia recorrida de viaje. Así que, el objetivo es encontrar una secuencia de ruta que minimice el costo total de la ruta diseñada y los viajes adicionales hacía y desde el depósito.

Los modelos estocásticos y dinámicos son fundamentales para el diseño de sistemas de soporte de decisiones que respondan a cambios en las condiciones observadas a menudo en aplicaciones prácticas (Powell, 1995). La presente investigación muestra Un Algoritmo Rollout (RA) para la solución del problema de ruteo de único vehículo con demanda estocástica, desde una perspectiva de

[3] YANG, Wen-Huei; MATHUR, Kamlesh y BALLOU, Ronald. Stochastic vehicle routing problem with restocking. En: Transportation science. Vol. 34, No. 1, (2000); p 99-102.

reoptimización por medio de la programación dinámica. Este proyecto asume un algoritmo de solución de programación dinámica para el problema de enrutamiento con demandas estocásticas de un solo vehículo.

2. JUSTIFICACIÓN DEL PROYECTO

En las últimas décadas se ha incrementado la necesidad de satisfacer los requerimientos del cliente, no solo ofreciendo un producto de calidad, sino, también una entrega oportuna de sus exigencias. El problema de ruteo de vehículos con demanda estocástica busca optimizar el uso de los recursos para reducir los costos de recoger o dejar mercancías y mejorar la eficiencia utilizando tecnologías de computación; es de anotar que la demanda de un cliente se puede conocer antes de llevar a cabo el recorrido, sin embargo, hay situaciones donde solo se conoce en el momento en que el vehículo llega a la localización del cliente. El VRPSD modela situaciones reales tales como, la entrega de los productos del petróleo, gases industriales (Chepuri y Homem-De Mello, 2005), y el aceite de calefacción (Dror , 1985). Otros VRPSD tienen su origen en la entrega de productos a las ciudades en emergencia (Dessouky, 2005), hospitales, restaurantes, máquinas expendedoras (Yang, 2000), y las sucursales bancarias. Demandas al azar también están presentes en la recogida de dinero (Laporte, 1989), paquetes (Markovic, 2005), los lodos, basuras y los materiales reciclados procedentes de bancos, viviendas y plantas industriales.

El VPRSD pertenece a la categoría de problemas NP-hard; en teoría de la complejidad computacional, los problemas NP contienen los problemas de decisión que son resueltos por una MT (*Máquina de Turing*) no determinista en tiempo polinómico y de ahí su nombre: **N**on-Deterministic **P**olynomial-time. Por otro lado, un problema NP-hard puede ser descrito como el contenido de los problemas de decisión que son al menos tan difíciles como un problema de **NP**. Para el problema de ruteo de único-vehículo con demanda estocástica, es necesario usar algoritmos aproximados desde una perspectiva de reoptimización; como el *Algoritmo Rollout* (RA), el cual se extiende mediante la implementación de diferentes secuencias de base (solución a priori), políticas look-ahead y esquemas de pruning[4]. Algunos autores que han estudiado e implementado el *Algoritmo Rollout* (RA) para la solución de este tipo de problemas son: Bertsekas y Secomandi[5], los cuales propusieron e implementaron una solución para el VRPSD dinámico.

El VRPSD por ser un tema de aplicación en el día a día, se convierte en un tema atractivo de investigación. En el presente proyecto se programará el *Algoritmo Rollout* (RA), para obtener una solución al VRPSD, el cual se utilizará en la ampliación de fuentes académicas y brindar bases de conocimientos en logística e investigación de operaciones, así mismo, para la comparación con otros estudios que se realizaron y/o realizarán en el grupo de investigación OPALO, de la Escuela Estudios Industriales y Empresariales.

---

[4] NOVOA, Clara y STORER, Robert. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. En: European journal of operational research. Vol. 196, No 1 (2009); p 509-515.
[5] Ibid., p. 511.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Diseñar un aplicativo para la solución de problemas de ruteo de vehículos con demandas estocásticas (VRPSD) a través de Algoritmo Rollout (RA).

3.2 OBJETIVOS ESPECÍFICOS

- Aplicar el *Algoritmo Rollout* (RA) como método heurístico eficiente para la solución de VRPSD de acuerdo a la revisión bibliográfica previamente realizada.

- Construir el banco de prueba para validar el *Algoritmo Rollout* (RA).

- Desarrollar el software que de soluciones factibles al modelo.

- Validar la solución obtenida por el software para medir la calidad de los resultados con base en parámetros definidos anteriormente.

4. ALCANCE

Los resultados esperados al finalizar la investigación del presente proyecto de acuerdo a los objetivos planteados, son los siguientes:

- Presentar un plan que muestre un estado del arte de toda la información recopilada, un planteamiento del problema claro, las técnicas utilizadas para resolver el VRPSD, el algoritmo heurístico Rollout aplicado y los resultados arrojados al ejecutar la herramienta computacional.

- Implementar el algoritmo heurístico en lenguaje de programación y su respectiva validación mostrando las soluciones factibles al modelo.

- Realizar un artículo documentándose allí el análisis, resultados y conclusiones de la investigación hecha en dicho proyecto.

5. MARCO TEORICO

5.1 OPTIMIZACIÓN COMBINATORIA

La optimización combinatoria es una rama de la optimización en matemáticas aplicadas y la ciencia de la computación, la cual estudia los problemas que se identifican por presentar una cantidad finita de soluciones factibles y trabajar con variables discretas.

Los nuevos estudios de la optimización combinatoria se deben a la llegada de la computadora digital moderna. La mayoría de los métodos actualmente aceptados en la solución de los problemas de optimización combinatoria difícilmente se habrían tomado en serio hace 25 años, por la sencilla razón de que nadie había podido llevar a cabo los cálculos necesarios[6].

El problema de optimización combinatoria se puede expresar en la siguiente manera[7]:

Minimizar o maximizar

$$\boldsymbol{F}\ \ \boldsymbol{s}\ \ ;\ \forall\ \boldsymbol{s}\ \in \boldsymbol{S}\ en\ \boldsymbol{Z^E}\quad \boldsymbol{Z}\ es\ el\ conjunto\ de\ \text{números}\ enteros$$

Donde el conjunto $\boldsymbol{E}\colon\{\mathbf{1}, \mathbf{2}, \mathbf{3}, \dots, \boldsymbol{n}\}$

El conjunto de soluciones factibles $\boldsymbol{S}$ es un conjunto del subconjunto $\boldsymbol{Z^E}$**.**

Una función objetivo definida en $\boldsymbol{Z^E}\ \boldsymbol{a}\ \boldsymbol{R}$

[6] LAWLER, Eugene. Combinatorial Optimization: Networks and Matroids. [en línea]. Courier Dover, (1976); p 1-4. [Consultado 7 Jun. 2013]. Disponible en: <http://www.plouffe.fr/simon/math/CombinatorialOptimization.pdf>.

[7] GÓMEZ, David y RANGEL, Carlos. Formular las Metaheurísticas Búsqueda Tabú y Recocido Simulado para la solución del CVRP (Capacitated Vehicle Routing Problem). [en línea]. Bucaramanga (2011); 150 h. Trabajo de grado (Ingeniería Industrial). Universidad Industrial de Santander. Facultad de Ingeniería Físico-Mecánicas. [Consultado 7 Jun. 2013]. Disponible en: <http://tangara.uis.edu.co/biblioweb/tesis/2011/137753.pdf>

El problema de optimización consiste en buscar una solución

$$s^* \in S \; tal\ que\ F(s^*) < F(s) \quad para\ un\ problema\ de\ minimización$$

$$s^* \in S \; tal\ que\ F(s^*) > F(s) \quad para\ un\ problema\ de\ maximización$$

Dónde:

$F(s)$ → Representa la función objetivo y mide la calidad de las decisiones

S → Es el conjunto de soluciones factibles que satisfacen ciertas restricciones del problema.

El objetivo entonces es, encontrar el máximo o el mínimo de una determinada función sobre un conjunto finito de soluciones.

5.1.1 Complejidad computacional. La complejidad computacional estudia los recursos necesarios para resolver un problema como son el tiempo (número de pasos de ejecución de un algoritmo para resolver un problema) y el espacio (cantidad de memoria utilizada para resolver un problema), la complejidad depende de la dificultad del cálculo que está directamente relacionada con el uso de los recursos. Así mismo, la complejidad está determinada por un modelo de computación (generalmente la máquina de Turing).

Máquina de Turing

En 1935, el matemático y lógico inglés Alan Mathison Turing (1912-1954) se interesó en el problema de decisión de Hilbert[8], que preguntaba si podría existir un método general aplicable a cualquier enunciado, para determinar si éste era

[8] CORTÉZ, Augusto. Teoría de la complejidad computacional y teoría de la computabilidad. En: Rev. investig. sist. inform. Facultad de Ingeniería de Sistemas e Informática. Universidad Nacional Mayor de San Marcos. [en línea]. [consultado 8 junio 2013]. (2004); p. 102-105. Disponible en: <http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/risi/n1_2004/a14.pdf>

verdadero. El enfoque de Turing lo llevó a pensar lo que ahora se conoce como la máquina de Turing. La máquina de Turing es una entidad matemática abstracta que formalizó el concepto de algoritmo y resulto ser la precursora de las computadoras digitales. Con la ayuda de su máquina, Turing pudo demostrar que existen problemas irresolubles, que ninguna máquina de Turing (y, por ende, ningún computador u ordenador) será capaz de obtener su solución[9].

La máquina de Turing puede ser de dos tipos:

- Determinista: Cuando existe a lo sumo (como máximo) una posibilidad de ejecución.
- No determinista: Cuando existe al menos un par con más de una posible combinación de actuaciones, es decir, presenta varias alternativas de ejecución.

5.1.2 Clases de complejidad. La teoría de la complejidad computacional trata de clasificar los problemas que pueden, o no pueden ser resueltos con una cantidad determinada de recursos. Las dos clases son P y NP, estas relacionan sus siglas con el modo de computación de la máquina de Turing.

- **Complejidad P**. Conjunto de problemas de decisión que pueden ser resueltos en tiempo polinomial respecto al tamaño de la instancia por una máquina secuencial determinista.

- **Complejidad NP (Non-Deterministic Polynomial Time).** La clase NP contiene aquellos problemas que pueden ser resueltos en tiempo polinomial por una máquina de Turing no determinista. O lo que es lo mismo, se puede verificar la solución en tiempo polinomial por una máquina secuencial no determinista.

[9] ALFONSECA, Manuel. La máquina de Turing. [en línea]. p 165-168 [Consultado 7 Jun. 2013]. Disponible en: <http://www.sinewton.org/numeros/numeros/43-44/Articulo33.pdf>

- **NP-completo (Non-Deterministic Polynomial Time- Complete).** Un problema P es NP completo si P ∈ NP y todos los problemas de clase NP pueden ser reducidos a un problema P en un tiempo polinomial, esto implica que son problemas dentro de la clase NP difíciles de resolver[10].

- **NP-hard (Non-Deterministic Polynomial Time-Hard).** Estos problemas son parcialmente similares pero más difíciles que los problemas NP-completo y necesitan en su mayoría tiempo exponencial para ser solucionados.

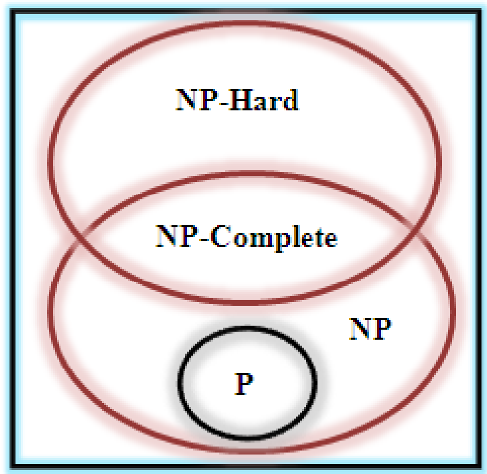
Para resumir algunas relaciones se tiene que:

- NP completo ⊆ NP hard
- NP completo ⊆ NP
- P ⊆ NP

Las relaciones anteriores se ilustran a continuación en la figura 1.

[10] HAMALAINEN, Wilhemeiina. Class NP, NP-complete, and NP-hard problems. [en línea]. (2006). [Consultado 7 Jun. 2013]. Disponible en: <https://cs.joensuu.fi/pages/whamalai/daa/npsession.pdf>

Figura 1. Relaciones existentes entre las clases



5.2 PROBLEMA DE RUTEO DE VEHÍCULO (VRP)

Uno de los problemas de optimización combinatoria más populares, es el problema de ruteo de vehículos *(Vehicle Routing Problem, VRP)*, su estudio ha generado técnicas de solución exactas y heurísticas de aplicación general. El VRP consiste en una serie de clientes que se encuentran geográficamente dispersos donde cada cliente presenta una demanda, se tiene un depósito y una flota de vehículos con capacidad limitada Q, el objetivo del problema es "encontrar un conjunto de rutas con costo mínimo visitando a cada uno de los clientes y las demandas sean satisfechas sin violar las restricciones de la capacidad del vehículo"[11]. Para reflejar mejor la incertidumbre que existe en nuestra vida diaria, se asume un problema modelado por variables aleatorias tales como el conjunto de clientes a visitar, el tiempo de viaje entre los clientes y las demandas de los clientes, haciendo parte de unos parámetros no determinísticos, clasificándose en

[11] NOVOA, Clara., et al. A Set-Partitioning-Based Model for the Stochastic Vehicle Routing Problem. En: Computational Optimization Research At Lehigh:Technical Report. [en línea]. [consultado 12 junio 2013]. (2006); Disponible en: <http://www.optimization-online.org/DB_FILE/2006/12/1542.pdf>

la literatura como problema de ruteo de vehículos estocástico *(Stochastic Vehicle Routing Problem, SVRP).*

El VRP ha sido un tema muy estudiado a lo largo de la historia, debido a su importancia teórica y práctica. En general el diseño de rutas de vehículos es una de las funciones operativas más críticas en temas de transporte y de gran importancia para una eficiente gestión de la cadena de abastecimiento. El artículo de **Dantzig, Fulkerson y Johnson (1954)**, “es el primer registro en la literatura del VRP, estudiaron un TSP a escala relativamente grande y se propuso un método de solución”[12].

En la década de los 70’s se realizaron diversos estudios y surgieron otras versiones del VRP, por ejemplo, los modelos de flota de enrutamiento y la programación de los problemas de los sistemas de transporte expuestos en el artículo de **Levin (1971)**.

En la década de los 80´s se genera un “estancamiento” debido a que los problemas exigían una complejidad computacional para ser resueltos, lo cual la tecnología de la época no ayudaba. “Para una investigación más a fondo en los principales tipos de problemas, formulaciones y métodos de solución de esta era, se pueden consultar las obras de **Laporte y Norbert (1987), Assad (1988) y Laporte (1992)**”**[13].**

La década de los 90´s fue de gran importancia debido a los estudios que se realizaron por el avance tecnológico, la mejora en los medios de comunicación, la capacidad de los microordenadores y almacenamiento de datos, entonces el VRP dinámico (DVRP) comenzó a ser objeto de estudio y a ser un tema más frecuente en la literatura en mitad de los años de 1990. **Gendreau et al. (1998)** estudia las aplicaciones del VRP con meta-heurísticas tales como: (1) el recocido simulado,

[12] EKSIOGLU, Burak; VOLKAN, Arif Vural y REISMAN, Arnold. The vehicle routing problem: A taxonomic review. En: Computers & Industrial Engineering. Vol. 57, (2009); p, 1473.
[13] EKSIOGLU, VOLKAN y REISMAN, Op.cit., p1473.

(2) recocido determinista, (3) La búsqueda tabú, (4) los algoritmos genéticos, (5) los sistemas de hormigas, y (6) redes neuronales.

Para una investigación más profunda en el problema de ruteo de vehículo dinámico se puede remitir a estudios por **Psaraftis (1995)**.

5.2.1 Definición del TSP[14]. TSP puede ser definido como sigue: "Un agente es requerido para visitar cada $\boldsymbol{n}$ ciudades una vez y solo una vez, empezando por cualquier ciudad y retornando al lugar original de partida. ¿Qué ruta debería seleccionarse con el fin de minimizar la distancia total recorrida?". En lugar de la distancia, otras nociones tales como tiempo, costo, etc., pueden ser considerados también. Matemáticamente, el problema puede denotarse en las siguientes dos formas equivalentes:

1) Dado una "matriz de costo" $\boldsymbol{D} = (\boldsymbol{d_{ij}})$, donde $(\boldsymbol{d_{ij}}) =$ coste de pasar de la ciudad $\boldsymbol{i}$ a la ciudad $\boldsymbol{j}, (\boldsymbol{i}, \boldsymbol{j} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{n})$, encontrar una permutación $\boldsymbol{P} = (\boldsymbol{i_1}, \boldsymbol{i_2}, \boldsymbol{i_3}, \dots, \boldsymbol{i_n})$ del número entero $\boldsymbol{1}$ hasta $\boldsymbol{n}$ que minimice la cantidad $\boldsymbol{d_{i_1 i_2}} + \boldsymbol{d_{i_2 i_3}} + \cdots + \boldsymbol{d_{i_n i_1}}$. Esto quiere decir que cada solución es una permutación del conjunto de vértices $(\boldsymbol{i}, \boldsymbol{j} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{n})$, en total se tendrían $\boldsymbol{n}$! Soluciones factibles posibles.

2) Dado una "matriz de costo" $\boldsymbol{D}$ como arriba, determinar $\boldsymbol{x_{ij}}$ como minimización de la cantidad $\boldsymbol{Q} = \quad_{ij} \boldsymbol{d_{ij} x_{ij}}$ sujeto a

- $\boldsymbol{x_{ii} = 0}$
- $\boldsymbol{x_{ij} \epsilon \;\; 0, 1}$
- $\quad_{i} \boldsymbol{x_{ij}} = \quad_{j} \boldsymbol{x_{ij}} = \boldsymbol{1}$
- Para cualquier subconjunto $\boldsymbol{S} = \{\boldsymbol{i_1}, \boldsymbol{i_2}, \dots, \boldsymbol{i_r}\}$ del número entero $\boldsymbol{1}$ a $\boldsymbol{n}$**,**

[14] SHEN, Lin. Computer Solutions of the Traveling Salesman Problem. [en línea]. (Agosto 18, 1965). [Consultado 3 Jun. 2013]. Disponible en: <http://www3.alcatel-lucent.com/bstj/vol44-1965/articles/bstj44-10-2245.pdf>

$$x_{i_1 i_2} + x_{i_2 i_3} \ldots + \; x_{i_{r-1} i_r} + x_{i_r i_1} \begin{array}{l} < \boldsymbol{r \; para \; r < n} \\ \leq \boldsymbol{n \; para \; r = n} \end{array}$$

La segunda versión es la formulación del problema del agente viajero como programación lineal, por lo tanto el problema puede ser resuelto como tal.

5.2.2 Definición del VRP. El problema de ruteo de vehículos se formula mediante la teoría de grafos: Sea el grafo $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A})$, donde $\boldsymbol{V} = \{\boldsymbol{v_0}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_n}\}$ es un conjunto de vértices que representan a los clientes y $\boldsymbol{A} = \{ \; \boldsymbol{v_i}, \boldsymbol{v_j} \; : \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{0}, \ldots, \boldsymbol{n} \;, \boldsymbol{v_i}, \boldsymbol{v_j} \in \boldsymbol{V}\}$ es el conjunto de arcos que representan caminos entre dos clientes $\boldsymbol{i}$ y $\boldsymbol{j}$. Asociado al grafo se define una matriz $\boldsymbol{L} = (\boldsymbol{L_{ij}})$, donde el coeficiente $\boldsymbol{L_{ij}}$ define la distancia entre los nodos $\boldsymbol{v_i}$ y $\boldsymbol{v_j}$, y son proporcionales al costo del viaje correspondiente al arco. Para satisfacer la demanda de los clientes se dispone de una flota de vehículos con capacidad de carga $\boldsymbol{Q}$. Cada vehículo parte del nodo depósito $\boldsymbol{v_0}$, se realiza las visitas a cada uno de los clientes y regresa de nuevo al depósito, naturalmente después de satisfacer la demanda de cada uno de ellos. "[15] Una matriz simétrica de costos de viaje sería $\boldsymbol{C} = \boldsymbol{C_{i,j}}$, definida en *A.* En la Figura 2 se ilustra un ejemplo de ruteo de vehículo.

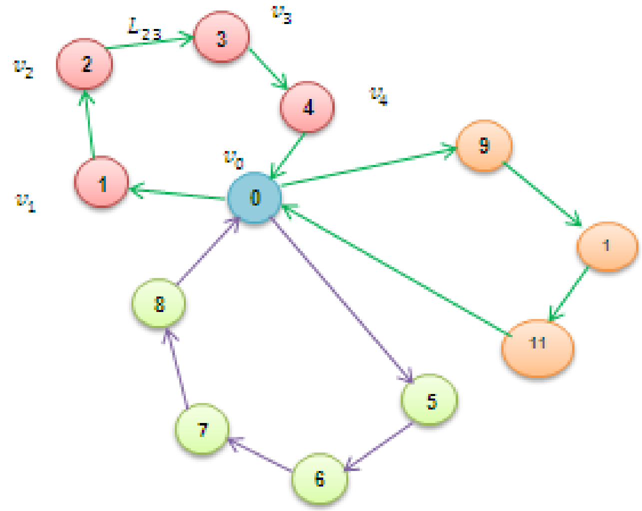
El objetivo del VRP es obtener un conjunto de rutas para una flota de vehículos en donde cada ruta comienza y termina en el depósito con un costo mínimo de viaje, de tal forma que los clientes son visitados una sola vez exactamente por un solo vehículo y la capacidad de estos no puede excederse. [16]

Figura 2. Solución del VRP con 1 depósito, 11 clientes y 2 vehículos.

---

[15] CHEPURI, Krishana y HOMEM-DE-MELLO Tito. Solving the Vehicle Routing Problem with Stochastic Demands using the Cross Entropy Method. [en línea].(2003). [Consultado 3 Jun. 2013]. Disponible en: <http://www.optimization-online.org/DB_FILE/2004/06/893.pdf>.

[16] GOODSON, Justin C; THOMAS Barrett W y OHLMANN Jeffrey W. Restocking-Based Rollout Policies for the Vehicle Routing Problem with Stochastic Demand and Duration. En: [en línea]. (mar. 2009). [Consultado 3 Jun. 2013]. Disponible en <http://www.slu.edu/~goodson/papers/GoodsonRestockingVRPSDL.pdf>.



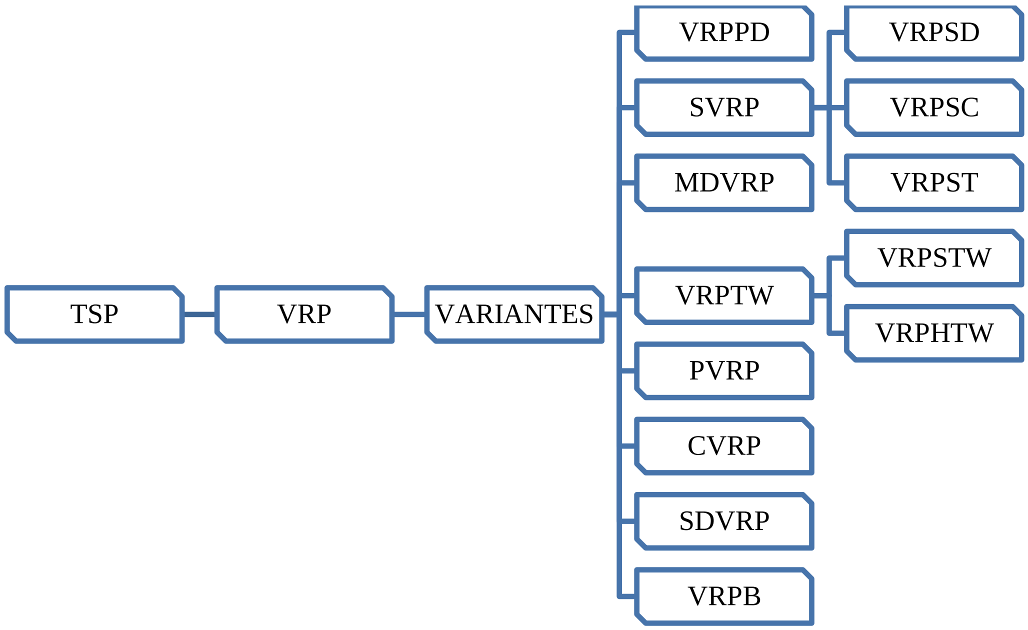
En la práctica, las exigencias de los clientes se dan a conocer con certeza cuando el vehículo llega a la locación del cliente, puede ocurrir también que la capacidad del vehículo se exceda y este se ve obligado a regresar al depósito para reabastecerse y regresar a su viaje para suplir las necesidades de los clientes faltantes, esta clase de eventualidades hacen que el VRP se vuelva más complejo y más difícil de dar una solución.

5.2.3 Variantes del VRP. Diferentes variaciones del VRP se han propuesto con el ánimo de acercarse a contextos reales del problema[17].

Las variantes más importantes se muestran en la figura 3.

[17] GOMEZ ATUESTA, David Fernando y RANGEL CARVAJAL, Carlos Eduardo. Formular las metahurísticas búsqueda tabú y recocido simulado para la solución del CVRP (capacitated vehicle routing problema).Bucaramanga, 2011, 150p. Tesis (Ingeniero Industrial). Universidad Industrial De Santander. Facultad de Ingeniería Fisico-Mecanicas. Escuela de Estudios industriales y Empresariales.

Figura 3. Variaciones VRP



Fuente: Adaptada a partir de GÓMEZ, Davis y RANGEL, Carlos. Formular las Metaheurísticas Búsqueda Tabú y Recosido Simulado para la Solución del CVRP (Capacitated vehicle Routing Problem). Bucaramanga, 2011, 24p.

A continuación, se presenta cada una de las variantes del problema VRP[18]

- **Problema de Ruteo de Vehículos con Envío y Recogida (VRPPD).** Es una de las variaciones presentadas en problemas de ruteo de vehículos el cual se tiene un determinado producto que quiere ser llevado de un cliente i a un cliente j, o bien pueda presentarse que los clientes devuelvan los productos a la empresa proveedora. Un factor importante es que los vehiculos tenga la capacidad suficiente para embarcar los productos devueltos y que estos cumplan las rutas establecidas.

[18] ROCHA, Linda; GONZALEZ, Cristina y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011); p. 35-55.

- **Problema de Ruteo de Vehículo Probabilístico (SVRP).** Se tiene en cuenta la probabilidad de ocurrencia P_i en los problemas presentados del VRP.
- **Problema de Ruteo de Vehículos con Demandas Estocásticas (VRPSD)**[19]. En esta situación se presenta que las demandas son inciertas, solo se tiene una distribución de probabilidad conocida. En vista de la complejidad de esta clase de problemas, la mayor parte de la investigación puede asociarse a un enfoque de optimización a priori o como enfoque de reoptimización.
- **Problema de Ruteo de Vehículos con Clientes Estocásticos (VRPSC)[20].** Ocurre todo lo contrario que el caso anterior, los clientes son variables aleatorias con distribuciones de probabilidad conocida.
- **Problema de Ruteo de Vehículos con Tiempos Estocásticos (VRPST).** Aquí los tiempos de viaje y servicio son variables aleatorias.
- **Problema de Ruteo de Vehículos con Múltiples Depósitos (MDVRP).** Este problema presenta múltiples depósitos y flota de vehículos. El objetivo es minimizar costos y asignar un depósito a cada cliente.
- **Problema de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW).** En el problema de ruteo de vehículo, este caso presenta un intervalo de tiempo fijo durante el cual se puede visitar cada cliente, siendo esta una restricción adicional.
- **Problema de Ruteo de Vehículos con Ventanas de Tiempo Fuertes (VRPHTW)**. Para este problema se puede considerar ventanas de tiempo duras en las que no es posible realizar la entrega al cliente fuera de los intervalos de tiempo establecidos.
- **Problema de Ruteo de Vehículos con Ventanas de Tiempo Suaves (VRPSTW).** Se puede realizar las entregas a cada cliente fuera del intervalo de tiempo, pero con una penalización pertinente.

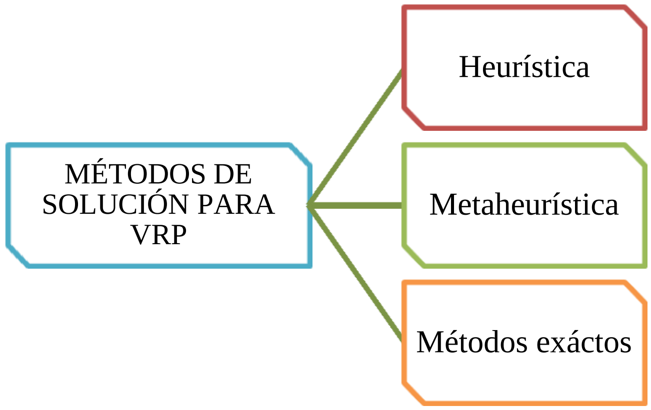
[19] NOVOA, Clara., et. al, Op.cit., p 3.
[20] Ibid.,p. 37.

- **Problema de Ruteo de Vehículos Periódicos (PVRP).** Es la variante del VRP con entregas realizadas sobre un periodo de M días, en lugar de un único día de servicio.
- **Problema de Ruteo de Vehículos Capacitados (CVRP)*.*** La flota de vehículos y la capacidad de cada uno de ellos es limitada, siendo esta la restricción del problema; también se debe tener en cuenta que la demanda total asignada no puede exceder la capacidad del vehículo.
- **Problema de Ruteo de Vehículos con Entregas Divididas (SDVRP).** Es la variante del VRP con clientes que pueden ser servidos por más de un vehículo.
- **Problema de Ruteo de Vehículos con Devoluciones (VRPB).** Los clientes pueden demandar o devolver artículos. Se debe tener en cuenta que las devoluciones de los clientes caben en el vehículo. Pero además, se debe cumplir que todas las entregas se realizan antes de las recogidas. Esto se debe al hecho de que los vehículos se cargan por la parte trasera y que la recolocación de la carga en los vehículos se considera antieconómica o no factible.

5.2.4 Métodos de solución para casos del VRP. Actualmente, los algoritmos para resolver las distintas instancias del VRP son muy variados en distintos aspectos, todas estas variantes del VRP tienen un grado de dificultad al momento de lograr su solución, entre más grande sea el tamaño del problema, el tiempo y el esfuerzo computacional requerido para resolverlo también aumenta exponencialmente; lo que lleva a usar métodos rápidos y eficientes para hallar soluciones aproximadas.

En lo que corresponde a los métodos de solución de los problemas VRP, se ha dividido en tres categorías y se pueden observar en la figura 4:

Figura 4. División según métodos de solución para el VRP



Fuentes: Adaptado a partir de ROCHA, Linda; GONZÁLEZ, Elsa y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011); p.42.

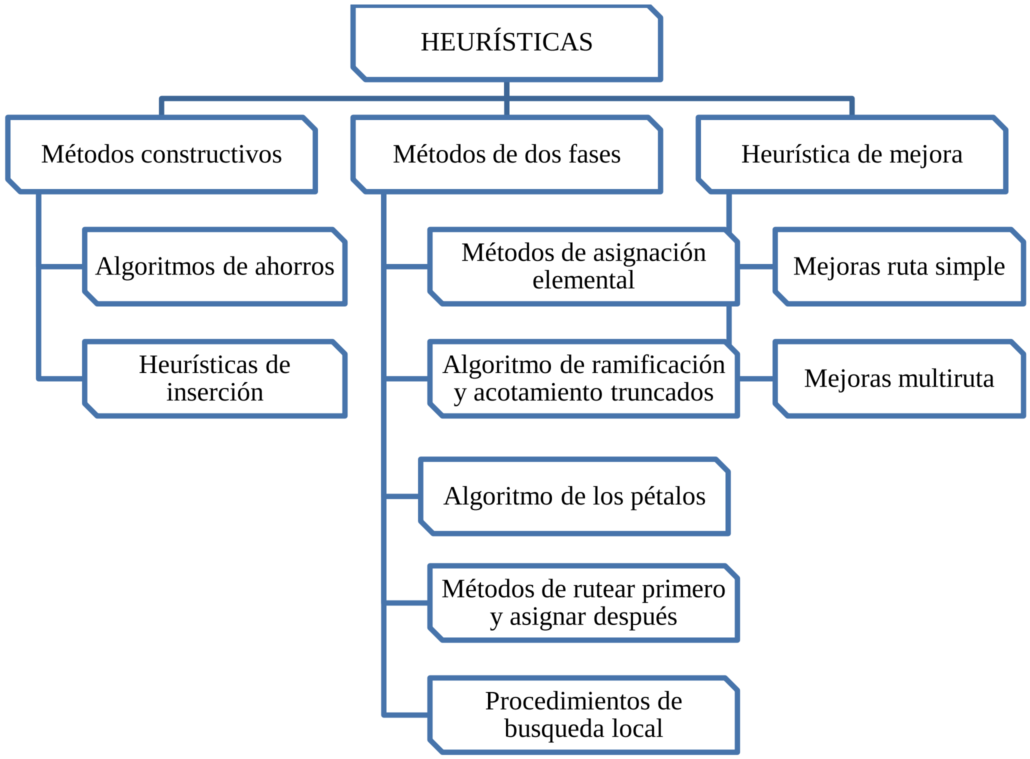
5.2.4.1 Heurística[21] Son procedimientos que permite obtener soluciones de buena calidad para un problema dado. Esto permite tener menores tiempos de ejecución, pero sin asegurar la optimalidad de la solución. Clarke y Wright[22], propusieron el primer algoritmo que resultó efectivo para resolver el VRP en 1964. La mayoría de las heurísticas clásicas para resolver el VRP fueron desarrolladas entre 1960 y 1990.

Dependiendo de cómo acometen su labor, las heurísticas (para el problema de rutas de vehículos) pueden clasificarse como muestra la figura 5:

[21] LÜER, Armin., et.al. El problema de rutas de vehículos: Extensiones y métodos de resolución, estado del arte. [en línea]. (2009). [Consultado 3 Jun. 2013]. Disponible en: <http://ceur-ws.org/Vol-558/Art_23.pdf>.

[22] RIBAS, Sabir., et.al. A hybrid algorithm for the Vehicle Routing Problem with Time Windows. [en línea]. (2011). [Consultado 3 Jun. 2013]. Disponible en: <http://www2.ic.uff.br/~satoru/conteudo/artigos/PAPER%20IESM2011-SABIR.pdf>

Figura 5. División para los métodos de solución heurísticos



Fuente: Adaptado a partir de ROCHA, Linda; GONZÁLEZ, Elsa y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011); p. 44.

a. Métodos Constructivos[23]. Esta clase de métodos se van elaborando a medida que progresan, es decir que no parten de una solución factible. Aquí encontramos el algoritmo de ahorro y las heurísticas de inserción, que son bastante conocidas en la parte de investigación de operativa. La figura 6 muestra a continuación la clasificación del algoritmo de ahorro.

- ***Algoritmo de ahorros propuesto por Clarke y Wright***, generalmente se aplica para problemas de ruteo en los cuales el número de vehículos son variables de decisión y al utilizar los arcos se obtiene el mayor ahorro en distancia. En la solución se puede encontrar que haya dos rutas diferentes y estas rutas puedan ser combinadas para obtener una nueva, esta nueva ruta

[23] ROCHA, GONZALEZ y ORJUELA, Op.cit.,p 38.

tiene mayor ahorro en sus arcos lo que generaría escogerla para solución óptima; este método genera muy buenas rutas al inicio del procedimiento, pero al final muestra rutas circulares.

- ***Mejoras del Algoritmo de Ahorros propuesto por Laporte, Toth y Vigo*** generalizaron los ahorros mediante un parámetro llamado Shape Parameter o Parámetro de Forma que penaliza la unión de rutas con clientes lejanos.

Figura 6. División para los métodos de solución mediante algoritmos de ahorro.

Fuente: Adaptado a partir de ROCHA, Linda; GONZÁLEZ, Elsa y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011); p. 45.

- Por su parte los ***Algoritmos de los Ahorros Basados en Matching*** es una nueva propuesta del algoritmo de ahorros estándar, el cual busca la unión de dos rutas pero tiene en cuenta las uniones sub-siguientes posibles. Aquí se puede lograr el ahorro con la combinación de las rutas correspondientes siempre y cuando sean factibles, esto se realiza mediante un gráfico, allí están todas las rutas posibles en nodos y en cada dos nodos se encuentra un arco.

En la **Heurística de inserción** se encuentra dos algoritmos de dos fases cada uno, que aplican a problemas con un número de vehículos no específico[24], a su vez este se clasifica en[25]:

- ***La heurística de inserción secuencial de Mole & Jameson*,** se tiene en cuenta ciertos parámetros para expandir una ruta en construcción. Para incluir un cliente existen dos medidas; la primera es el costo que tiene de incluir un cliente que no ha sido visitado en la ruta, la segunda es calcular el valor de ese cliente no visitado teniendo en cuenta las distancias sin reordenar los nodos que ya están en la ruta.

- ***La heurística de inserción en paralelo de Christofides, Mingozzi & Toth*,** utiliza dos parámetros controlados γ y μ , primeramente se utiliza la primera fase de la heurística anterior (algoritmo de Mole & Jameson) y así se logra obtener rutas compactas conservando los clientes iniciales de cada ruta junto con las rutas finales; luego en la segunda fase se crean las rutas y se inserta el resto de los clientes en ellas.

b. Método de Dos Fases

- ***Métodos de Agrupamiento Elemental*,** hace parte de este el algoritmo de barrido, el algoritmo basado en asignación generalizada y la heurística basada en localización.

- ***Ramificación y Acotamiento Truncados*,** se tienen tanto niveles como rutas de vehículos en el árbol de búsqueda y cada nivel posee un conjunto de rutas de vehículos. Christofides, Mingozzi y Toth proponen implementar y determinar

---

[24] LAPORTE, Gilbert; GENDREAU, Michel y HERTZ, Alain. An aproximation algorithm for the traveling salesman problem with time windows. En: Institute for Operation Research and de Management Science. Vol. 45, No. 4, (1998); p 639-641
[25] ROCHA, GONZALEZ y ORJUELA, Javier, Op.cit.,p 46.

una rama en cada nivel, descartando una ruta al paso de ir seleccionando la ruta. Se puede mantener pocas rutas en cada nivel, construyendo un árbol limitado.

- ***Algoritmos de los Pétalos*****,** este algoritmo es una extensión del algoritmo de barrido y se utiliza para generar varias rutas llamadas pétalos con el fin de hacer una selección final resolviendo un Set Partitioning Problem. Se tiene un conjunto de *R* rutas, allí cada cliente es visitado por varias rutas, se selecciona un sub-conjunto de *R* que visite una sola vez a cada cliente.

- ***Métodos de Ruteo Primero y Asignación Después*****,** consta de dos fases. Sin tener en cuenta las restricciones, primero se determina una gran ruta que visite a cada una de los clientes resolviendo así el TSP; segundo la gran ruta determinada se descompone en varias rutas factibles, en las cuales la solución de la primera y la capacidad del vehículo determinan la mejor partición.

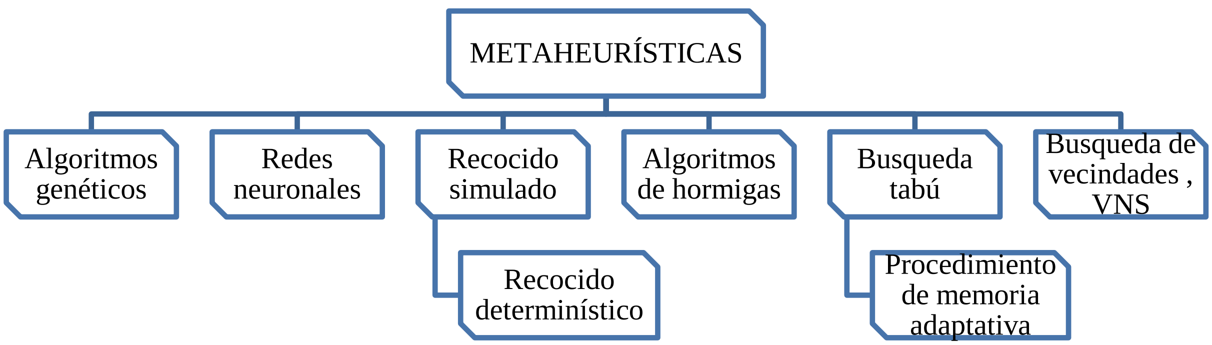
- ***Procedimientos de Búsqueda Local (Local Search Procedures).*** Aplicados para mejorar la solución que se ha obtenido. Se define un conjunto de soluciones vecinas partiendo de una solución primaria para que después se reemplace por una solución encontrada vecina de menor costo. Esto se aplica varias veces hasta que ya no se pueda mejorar más la solución.

5.2.4.2 Metaheurística[26]. Al haber una gran variedad de problemas en los cuales no existe un método o algoritmo confiable de resolución, ya sea por la complejidad del problema mismo o por falta de documentación y estudios para dar solución a este; la metaheurística es una táctica (heurística) general para esto.
Este algoritmo se basa en las observaciones de la naturaleza, la evolución biológica, procesos asociados a la manufactura, siendo importante en la

[26] LÜER, Armin., et.al, Op.cit.,p 4.

investigación de operaciones, ya que puede ser aplicado a problemas de optimización combinatoria con resultados muy cercanos al óptimo. Las principales metaheurísticas utilizadas se muestran en la figura 7:

Figura 7. División para los métodos de solución metaheurísticos



Fuente: Adaptado a partir de ROCHA, Linda; GONZÁLEZ, Elsa y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011);p. 48.

- ***Algoritmos Genéticos*****,** son algoritmos evolutivos que consiste en un conjunto de cromosomas, donde cada elemento de éste se le llama gen. Este conjunto hace parte de una población, en el cual cada iteración hace referencia a una generación, para generar nuevos individuos se aplican diversos factores, que son agregados a la población, en un proceso iterativo que trata de escapar de mínimos locales. Para resolver problemas de ruteo de vehículo han tenido bastante éxito, se ha trabajado también como algoritmos meméticos (algoritmos genéticos con algún procedimiento adicional de búsqueda local) para situaciones en las cuales la flota de vehículos es heterogénea. Los algoritmos genéticos han sido aplicados también en forma pura y en forma hibrida, mezclando características de otras metaheurísticas.

- ***Búsqueda en Vecindarios Variables*****,** denominada VNS; dada una solución inicial aleatoria, se usa un algoritmo de búsqueda local eficiente, vecindarios progresivamente más lejanos (y grandes) y a partir de esto se explora, en caso de que se encuentre una solución mejor, se realiza el recorrido hasta ella y de esta forma se reinicia la búsqueda de vecindarios de esta. Con el transcurso del tiempo, en este método de trayectoria nacen un conjunto de variantes para solucionar problemas demasiado grandes o lograr mejoras en la resolución de estos.

- ***Recocido Simulado*****,** consiste en que las estructuras cristalinas de un metal que es calentado a altas temperaturas y que luego es enfriado lentamente, se reorganicen en la configuración mínima de energía, logrando así asemejar el proceso de manufactura. Este método de trayectoria en la versión computacional, para realizar una analogía con la situación física, la temperatura es discretizada admitiendo de esta forma soluciones peores que la mejor encontrada, permitiendo escapar óptimos locales con una probabilidad proporcional a la distribución termodinámica de Boltzmann. Se permite hacer cambios como el patrón de enfriamiento o la probabilidad de aceptar una peor solución, con el fin de buscar mejoras y por consiguiente facilita la implementación en el problema.

- ***Búsqueda Tabú*****,** en la proximidad de la situación actual, se busca otra solución que mejore la evaluación de la función objetivo o alguna de sus características, las cuales son marcadas como tabú. En este método de trayectoria, el algoritmo escapa de óptimos locales y evita que entre en un ciclo.

- ***Colonias de Hormigas***[27], cuando la hormiga detecta un trayectoria para su fuente de alimento, ella deposita en el camino una sustancia llamada feromona, de acuerdo a la calidad del alimento y la longitud recorrida se determina la cantidad de feromona, las hormigas al no detectar presencia de feromonas actúan moviéndose aleatoriamente, pero si ocurre lo contrario ella decidirá irse por el camino que contenga más feromona y por consiguiente se aumentara más dicha sustancia en este trayecto, surgiendo así un comportamiento denominado autocatalítico, lo que hace que al haber más feromonas el trayecto se vuelve más atractivo para ellas. Este algoritmo simula el comportamiento de una colonia cuando busca su alimento, cada una de las hormigas realiza una trayectoria diferente, lo que indica que tan bueno puede ser tomar la trayectoria que muestran y la información histórica (bajo la forma de feromona) que dice que tan bueno fue tomar dicha decisión.

- ***Enjambre de Partículas***[28], con el fin de dar una solución eficiente, se simula una búsqueda realizada por entes colaborativos, teniendo en cuenta las interacciones entre ellos.

5.2.4.3 Métodos exactos[29]. Forman parte de modelos como programación lineal (enteros) y llegan a una solución factible gracias a algoritmos de acotamiento del conjunto de todas las posibles soluciones[30]. Su clasificación se muestra en la figura 8.

---

[27] OLIVERA Alfredo. Heurísticas para problemas de ruteo de vehículos. En: Instituto de computación, facultad de ingeniería, universidad de la república. [en línea]. Montevideo, Uruguay, Agosto 2004. [consultado 25 may. 2013]. Disponible en <http://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0408.pdf>

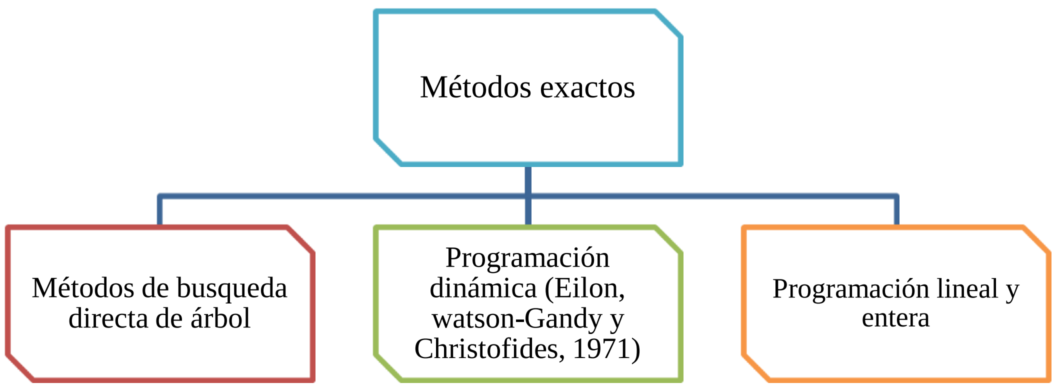
[28] LÜER, Armin., et.al, Op.cit., p 6.

[29] AZI, Nabila; GENDREAU, Michel y POTVIN Jean-Y ves. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. En: European Journal of Operational Research. Vol. 178, (2007); p. 756-763.

[30] ROCHA, Linda; GONZALEZ, Cristina y ORJUELA, Javier, Op.cit.,p 42

- ***Métodos de Búsqueda Directa de Árbol*****,** de acuerdo a ciertos criterios específicos se empieza una búsqueda sobre todos los nodos de un árbol. Existen métodos tales como Asignación de la Cota Inferior, Algoritmo de ramificación y corte, Algoritmo de ramificación y acotamiento, algoritmo de búsqueda de árbol.

Figura 8. División para los métodos exactos de solución



Fuente: Adaptado a partir de ROCHA, Linda; GONZÁLEZ, Elsa y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011); p. 42.

- ***Programación Dinámica*****,** propuesto por Eilon, Watson-Gandy y Christofides en 1971. Busca el costo mínimo al usar K vehículos, teniendo en cuenta la longitud de una ruta de vehículos a través de todos los vértices del sub-conjunto para la función del costo. Se debe tener en cuenta un número fijo de m vehículos para luego encontrar el costo correspondiente de todos los sub-conjuntos de vértices con esta cantidad de vehículos.

- ***Programación Lineal y Entera*****,** escoge mediante un modelo de optimización las soluciones que serán consideradas en un conjunto de soluciones factibles previamente definidas.

5.3 PROBLEMA DE RUTEO DE VEHÍCULO ESTOCÁSTICO (SVRP)

En la gestión de la cadena de suministro uno de los factores importantes es la coordinación de la logística de operaciones debido a que es un proceso integrado y debe desarrollarse correctamente para así ofrecer velocidad de respuesta al mercado, ofreciendo valor al cliente y a su vez generar costos mínimos. Dentro de estos sistemas la mayoría de veces los valores son variables desconocidas, las más comunes son[31]:

- *Los clientes estocásticos:* El cliente i está presente con una probabilidad de p_i y ausente con una probabilidad $1 - p_i$.
- *Demandas estocásticas:* La demanda del cliente es una variable aleatoria.
- *Tiempos estocásticos:* El servicio del tiempo del cliente y el tiempo de viaje de un cliente a otro, son variables aleatorias.

Normalmente el SVPR se modela en dos etapas: la primera es una solución a priori en el que las variables aleatorias se calculan, teniendo en cuenta distribuciones de probabilidad; la segunda etapa es hacer uso de un recurso que se aplica como acción correctiva a la primera etapa, generando un costo adicional o un ahorro que se tendrá en cuenta en el próximo diseño de la primera etapa.

Se ha estudiado esta clase de problemas en dos marcos[32]: Programa de Programación Restringida *(Chance Constrained Program, CCP)* y Programación Estocástica con Recurso *(Stochastic Program with Recourse, SPR)*. En el CCP el problema consiste en diseñar una serie de rutas en el que las probabilidades de falla están restringidas a estar por debajo de cierto umbral especificado, reduciendo así el problema a un VRP determinista. El objetivo es controlar la

[31] CORDEAU, Jean François., et al. Vehicle Routing. En: Handbooks in Operations Research & Management Science: Transportation. 2007. p.367.
[32] C.Y. Cheong, et al. A multiobjetive evolutionary algorithm for solving vehicle routing problem with stochastic demand. En: Congreso sobre computación evolutiva (16-21, Julio, 2006: Vancouver, Canadá). Memorias. Instituto de ingenieros eléctricos y electrónicos- IEEE, 2006. p.1370-1377.

probabilidad del fracaso de la ruta, ignorando el costo que estos generan. El SPR considera las distribuciones de demanda de los clientes y trata de disminuir el costo esperado del transporte, en el que se incluyen el costo de viaje más el costo que generan las políticas de recurso. El SPR es más completo de resolver que el CCP.

5.3.1 Variantes del SVRP. La incertidumbre se encuentra presente en diferentes partes del problema de enrutamiento de vehículos, los cuales podemos dividir en: VRP con clientes estocásticos (VRPSC), VRP con demandas estocásticas (VRPSD), VRP con el tiempo de viaje estocástico (VRPSTT) y el VRP con clientes y demandas estocásticos (VRPSCD).

- **VRP con clientes estocásticos (VRPSC).** Se presenta cuando los clientes con determinadas demandas, tienen una probabilidad $\boldsymbol{p_i}$ de estar presente. Una solución de la primera etapa consiste en una ruta para los vehículos que visitan al depósito y al conjunto de clientes una sola vez, el conjunto de clientes ausentes es revelado y la solución de la segunda etapa consiste en seguir las rutas de la primera etapa, mientras que se saltan los clientes ausentes.

- **VRP con demandas estocásticas (VRPSD).** Ocurre cuando las exigencias o demandas de la entrega individual se comportan como variables aleatorias. Tillman en 1969 fue el primero en proponer este problema en un contexto multidepósito, proponiendo una solución heurística de ahorro.

- **VRP con el tiempo de viaje estocástico (VRPST)**. Se describe el entorno de incertidumbre, el estado del tráfico de la carretera. Los vehículos siguen rutas planificadas, y se puede generar una multa cuando la duración de estos es mayor a la del plazo determinado. Kao en 1978 propuso una heurística basada en la programación dinámica y en la enumeración implícita para el TSP con el tiempo de viaje estocástico.

- **VRP con clientes y demandas estocásticos (VRPSCD**). Es una combinación del VRPSC y el VRPSD. En la primera etapa se construye un modelo de ruta en el que los clientes se visitan una sola vez, iniciando y terminando su recorrido en el depósito. Durante este recorrido, los clientes ausentes se van revelando con una demanda de cero y la demanda de los demás clientes se conoce únicamente cuando el vehículo llega. En la segunda etapa, se soluciona siguiendo la primera etapa a excepción de que los clientes ausentes, no se tienen en cuenta y cuando la capacidad del vehículo se excede, este tiene que regresar al depósito a reabastecerse y continua con su ruta empezando con el último cliente visitado, además se tiene que tener en cuenta que cuando la capacidad del vehículo se completa exactamente, este también regresa al depósito y reanuda su recorrido en el siguiente cliente en la ruta ya planeada.

5.4 PROBLEMA DE RUTEO DE VEHÍCULO CON DEMANDA ESTOCÁSTICA (VRPSD)

El VRPSD se da cuando en un VRP las demandas de los clientes son variables estocásticas o aleatorias, éstas siguen una distribución de probabilidad conocida y las exigencias reales del cliente solo se conocen al llegar al sitio. Si se presenta una ruta "fallida" debido a que la demanda del cliente ha excedido la capacidad actual del vehículo o éste ha quedado sin existencias, la distancia recorrida se incrementa, debido naturalmente a que le vehículo se ve obligado a regresar al depósito.

El objetivo del VRPSD es minimizar el costo total generado por la ruta prevista (a priori) y los viajes nuevos que se realizan hacia y desde el depósito. Para minimizar aún más los costos, Bertsimas et al. (1995) [33] diseña rutas a priori que puede determinar si regresa al depósito antes de que la capacidad del vehículo esté agotada, esta acción es denomina *Retorno Proactivo*. VRPSD ocurre tanto en situaciones de entrega y como en recogida.[34]

5.4.1 Definición del VRPSD.

En una determinada situación, se presenta que se debe distribuir un determinado producto desde una planta a un conjunto de clientes localizados geográficamente, utilizando un solo vehículo con cierta capacidad. El vehículo debe visitar a todos los clientes periódicamente para suministrar el producto y así reponer a su vez el inventario.

Las exigencias de cada cliente están determinadas por cada periodo y se modelan como independientes y con distribución de variables aleatorias conocidas. El vehículo se establece con una capacidad fija y no tiene conocimiento de las demandas que se encontrará en el transcurso de la ruta, por tanto existe una probabilidad positiva de que el vehículo se quede sin productos, generándose así una ruta de fallo, estos fallos producen sanciones que son los recurso que se utilizan a lo largo de la ruta, determinándose así la función de costos como la ruta total recorrida por el vehículo incluyendo los arcos visitados y las sanciones impuestas durante el recorrido. El objetivo es encontrar la ruta para la que el valor esperado del costo de la función sea mínimo en comparación con todas las otras rutas.[35]

Se define este problema como un grafo completo $G = (V, A, D)$, donde:

- $\boldsymbol{V} \rightarrow Conjunto\ de\ clientes\ \ nodos$.
  $denominando\ el\ nodo\ cero\ como\ el\ depósito, donde\ \boldsymbol{V} = \ \ \boldsymbol{0}, \boldsymbol{1}, \dots, \boldsymbol{n}$

[33] BERTSIMAS, Dimitris; CHERVI, Philippe Y PETERSON, Michael. Computational Approaches to Stochastic Vehicle Routing Problems. En: Transportation Science. Vol. 29, No. 4 (1995); p. 342–352.

[34] NOVOA, Clara. The Vehicle Routing Problem with Stochastic Demands Combining Static and Dynamic Approaches. p 1-7.

[35] CHEPURI, Krishana y HOMEM-DE-MELLO Tito, Op.cit., p 3.

- $\boldsymbol{A} \rightarrow Conjunto\ de\ arcos\ que\ une\ a\ los\ clientes, donde\ \boldsymbol{A} = \quad \boldsymbol{i}, \boldsymbol{j}\ : \boldsymbol{i}, \boldsymbol{j}\ \in \boldsymbol{V}, \boldsymbol{i} \neq \boldsymbol{j}$
- $\boldsymbol{D} \rightarrow Costos\ de\ viaje\quad distancia\ \ entre\ los\ clientes, donde\ \boldsymbol{D} = \ \ \boldsymbol{d_{ij}}: \boldsymbol{i}, \boldsymbol{j}\ \in \boldsymbol{V}, \boldsymbol{i} \neq \boldsymbol{j}$
- $\boldsymbol{\xi_i}, \ \rightarrow Demanda\ de\ los\ clientes, son\ variables\ estocáticas\ \boldsymbol{\xi_i}, \boldsymbol{i} = \boldsymbol{1}, \dots, \boldsymbol{n}$ $con\ distribuciones\ conocidas.$

La demanda real de cada cliente solo se conoce cuando el vehículo llega a la locación del cliente. También se supone que $\boldsymbol{\xi_i}$ no excede la capacidad del vehículo Q y sigue una distribución de probabilidad discreta $p_{ik} = Prob \quad \xi\mathrm{i} = \mathrm{k}\ \ , K = 0,1,2, \dots, K \leq Q$.

5.4.2 Políticas de servicio. Las principales investigaciones hechas sobre el VRPSD han estudiado tres enfoque principales que determina la política de servicio que abordará el vehículo para iniciar su ruta, enfoque de optimización a priori o política estática, enfoque de reoptimización o dinámico y mixta o política de descargue/abastecimiento preventivo.

A. Enfoque de optimización a priori: Este enfoque consiste en el diseño de rutas sin tener conocimiento de la demanda real del cliente, al momento del fracaso de la ruta, se adopta acciones de recursos sin necesidad de modificar el recorrido ya planificado. Se conoce también como política estática “de aquí y ahora”.

Siempre que se presente un fallo de ruta se van a requerir acciones de recursos, tales como:

- Se propone dos modelos sencillos que penalizan la función objetivo en caso de que se un fallo de ruta. **Stewart et.al.** en **1983** plantea que el primer modelo determine una sanción fija para cada ruta fracasada, mientras que el segundo modelo sanciona para cada unidad de demanda en exceso de la capacidad del vehículo[36]. Los dos modelos incluyen que los clientes deben ser visitados por la

[36] NOVOA, Clara., et al, Op.cit., p 4.

ruta, por tanto la función objetivo incluye el costo total de la ruta más el de la penalización.

- **Dror** y **Trudeau** en **1986** consideran que "después de un fallo de ruta, los clientes restantes sean visitados desde el deposito con entregas directas e individuales"[37].

- **Dror** en **1989** propone la *Estrategia de recurso tradicional* en que los vehículos se devuelven al depósito a descargar o reponer capacidad cuando la demanda del cliente no fue satisfecha, luego regresan para seguir con la ruta prevista.

- **Chepuri** y **de Mello** en el **2005** proponen un recurso de tipo penal en el que, en caso de que la ruta falle esta se termina hasta el final. Se penaliza evaluando la pérdida de ingresos y / o los gastos por no satisfacer al cliente. Es práctico utilizar este recurso cuando las entregas son de emergencias y no hay el tiempo suficiente para ir al depósito a reabastecerse[38].

- **Reabastecimiento preventivo***,* en esta estrategia los clientes son visitados en el orden de una ruta a priori y tienen que elegir de acuerdo a la demanda del cliente actual si procede al siguiente cliente o va al depósito a reabastecerse y de esta manera poder continuar su recorrido. "Ir al depósito a reponerse aún si el vehículo no está vacío o si la capacidad de este es mayor que la demanda esperada del próximo cliente, es conocido como *repoblación preventiva"*[39]*,* que consiste en evitar una mala situación en la que el vehículo no tenga la suficiente carga para atender un cliente y por lo tanto hay que realizar un viaje de ida y vuelta al depósito para completar la entrega al cliente.

[37] Ibid., p. 4.
[38] NOVOA, Clara., et al, Op.cit., p 2.
[39] BIANCHI, Leonora; et al. Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Parallel Problem Solving from Nature-PPSN. Enero, 2004. vol. 8, p. 450

- **Ak** y **Erera** en el **2006,** estudian una estrategia de recurso de vehículos dos a dos eliminando el supuesto de que estos operan independientemente de la ruta a priori. El estudio presenta dos vehículos en el cual uno de ellos sirve a una gran cantidad de demanda y otro a una cantidad más baja. "En caso de que la capacidad del primer vehículo sea superada por la demanda, el segundo vehículo agrega a cualquier cliente sin servicio al final de su recorrido. Si el segundo vehículo no está en condiciones de servir a todos los clientes adicionales en su ruta original, entonces la estrategia de *recurso tradicional* se utiliza para servir a estos clientes"[40].

- **Estrategia de recurso ampliada,** en esta estrategia, se permiten dos nuevas acciones de recurso cuando se da un fallo de ruta. "En primer lugar, un vehículo que ha completado su ruta original y tiene una capacidad restante puede atender a los clientes de las rutas fallidas directamente sin retornar al depósito. En segundo lugar, un vehículo que experimenta un fallo de ruta puede volver al depósito y luego realizar un *viaje extra* para atender la demanda de los clientes faltantes de rutas fracasadas"[41].

B. Enfoque de reoptimización: Ocurre siempre y cuando se den actualizaciones del sistema con el servicio, es decir, cada vez que el vehículo llegue a una ubicación y observa la demanda del cliente, el estado del sistema se actualiza. Esto da a entender que para resolver problemas VRPSD los datos no son conocidos completamente y esta información se revela a medida que pasa el tiempo, de manera dinámica, por consiguiente, se debe resolver el problema de manera secuencial.

---

[40] NOVOA, Clara., et al, Op.cit., p 5.
[41] NOVOA, Clara.,et al, Op.cit., p 6.

C. Enfoque mixto o política de descargue/abastecimiento preventivo:

Esta política combina elementos de ambas políticas estáticas y dinámicas, donde el vehículo sigue una ruta a priori, pero también está habilitada con reglas dependientes de estado que permiten reabastecimientos anticipados[42].

Los siguientes son ejemplos en un contexto real de ruteo de vehículo dinámico que han sido citados por Psaraftis[43], los cuales reflejan lo expuesto anteriormente:

- *Entrega de productos del petróleo, gases industriales y otros productos:*

El proveedor de petróleo debe organizar una serie de rutas para reponer los inventarios de un conjunto de clientes. La cantidad de aceite que necesita cada cliente no se conoce con precisión de antemano, y por lo tanto existe la posibilidad de que el tanque del camión que sirve al cliente no tenga suficiente aceite para responder totalmente a las necesidades del cliente. La posibilidad de falta de existencias es costosa y debe ser evitada, y también en este caso el envío de la flota de camiones se debe hacer de manera eficiente (de acuerdo con algún criterio).

- *Servicios de mensajería*:

Una mini furgoneta recorre un barrio de la ciudad recogiendo paquetes y paquetes exprés. Los pedidos de servicio llegan en tiempo real a la oficina central de despacho del correo y se transmite automáticamente ya sea por teléfono o por cualquier otro dispositivo a la minivan. La computadora a bordo de la camioneta procesa cada solicitud, mediante la inserción de manera apropiada en la secuencia de pedidos. Alternativamente, el procesamiento de cada solicitud se realiza de manera centralizada, ya sea por un ordenador central, o por un programador humano, que también decide que mini-van debe manejar cada petición.

[42] SECOMANDI, Nicola. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. En: Computers & Operations Research. Vol. 27. (2000); p. 1205.
[43] PSARAFTIS, Harilaos. Dynamic vehicle routing: Status and prospects. En: Operations Research. Vol. 61, No 1 (1995); p143-164

- *Recogida combinada y servicios de entrega:*

Aquí, una flota de vehículos está disponible a las solicitudes de demanda de servicios que llegan por teléfono o cualquier otro medio de comunicación (por ejemplo, una "cabina telefónica", ubicado en la calle) a un centro de expedición o programación central. En estas peticiones no se conocen de antemano al operador, sino que llegan en tiempo real, y debe ser reparado lo antes posible. El operador debe decidir en tiempo real qué vehículo debe enviar a una solicitud específica, y cómo modificar la ruta de ese vehículo (que puede comprometerse a recoger y / o dejar algunas otras peticiones). Al mismo tiempo, un nivel razonable de servicio debe mantenerse, y (con suerte) el uso de los recursos del vehículo debe ser "optimizados".

Se destacan dos políticas de servicio principal de **Dror**, **Laporte**, **Trudeau** en **1989**[44],

- **Entrega completa*:*** Se supone que cuando el vehículo llega a la locación del cliente debe servir la totalidad de su demanda, por eso es necesario en este caso hacer la siguiente suposición: $\xi i \leq Q(i = 1, \dots, n)$, de otra forma el problema no sería factible.

- **Entregas divididas:** Aquí la demanda de un cliente puede ser satisfecha por varios vehículos o puede ser dividida entre varias visitas de un mismo vehículo. En esta política la condición de que un cliente puede ser visitado una sola vez por un solo vehículo, es omitida. Se supone que $\xi i \leq Q(i = 1, \dots, n)$ no es necesario. Puede incluirse dentro de esta categoría la demanda dividida pero solo, en fracciones predefinidas.

[44] DROR, Mosher; LAPORTE, Gilbet y TRUDEAU, Pierre. Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks. En: Transportation Science. Agosto, 1989. Vol. 23, no.3,. p. 168.

Se da una clasificación de políticas de servicio de acuerdo al tiempo en el cual se conoce la demanda real del cliente propuesto por **Dror**, **Laporte**, **Trudeau** en **1989**[45],

- **Información anticipada completamente:** Aquí la demanda de los clientes son conocidas con anticipación antes de que las rutas sean planificadas. Lo que conduce al VRP.

- **Información de última:** La demanda del cliente se revela solo cuando el vehículo llega a su locación, SVRP. Se puede dar dos casos: el primero, es que la demanda sea conocida inmediatamente antes de la entrega o inmediatamente después de que el servicio sea realizado de forma completa.

5.4.3 Métodos de solución del VRPSD. Tillman (1969)[46] presento el primer trabajo de investigación sobre el VRPSD con múltiples depósitos, proponiendo un algoritmo para la solución basado en el método de ahorro de Clarke-Wright.

Golden y **Stewart** en **1978**[47] presentan *un modelo restringido* y *dos modelos de recurso.* En el primer modelo, se impone una penalización al exceso de capacidad del vehículo; en el segundo modelo, la penalización es proporcional a la demanda esperada que excede la capacidad del vehículo.

Dror y **Trudeau** **(1986)** desarrollaron más procedimientos los cuales muestran que "el costo de viaje esperado depende de la dirección de viaje del problema, incluso en casos simétricos"[48].

---

[45] DROR, LAPORTE y TRUDEAU, Op.cit., p 168.
[46] CHANG, Mei-Shiang. A vehicle routing problem with time windows and stochastic demands. En: Journal of the Chinese Institute of Engineers. Vol. 28, No. 5, (2005.); p.783-794.
[47] CORDEAU, Jean François., Op.cit., p 367.
[48] JUAN. A. Angel, et al. Applying Simulation and Reliability to Vehicle Routing Problems with Stochastic Demands. [en línea]. [consultado 25 May. 2013]. Disponible en <http://ceur-ws.org/Vol-589/paper02.pdf>.

La mayor contribución viene de **Bertsimas** en **1988**[49] que realiza un estudio e ilustra el método a priori con diferentes políticas de recursos (reoptimización) para resolver problemas derivados del VRPSD, resultados asintóticos y otras propiedades teóricas. Se analiza que en estas investigaciones la distribución de la demanda es binaria, es decir, el cliente i tiene una demanda única con probabilidad p_i o no tiene ninguna demanda con probabilidad $1 - p_i$.

Para tomar decisiones óptimas, el modelo de *decisión de Markov* ofrece el potencial suficiente para hacer uso de las acciones de recursos cada vez que una nueva información se revela, en lugar de depender de acciones de recurso estáticos. La desventaja más relevante del método es su uso en problemas de tamaño moderado y grandes por la cantidad excesiva de variables de decisión y de estado (problema de la maldición de la dimensionalidad), **Dror et al.** en **1989** y **Dror** en **1993**[50] realiza contribuciones de modelos teóricos dinámicos del VRPSD como proceso de decisión de Markov. **Dror et al.** en **1989**[51] plantea un modelo de un solo vehículo en el cual la decisión a tomar corresponde con el momento en que el vehículo llega a la ubicación del cliente y su demanda es revelada, tomando así dos posibles decisiones antes de servir al cliente: la primera sería no servir al cliente correspondiente y seguir a la siguiente locación, la segunda decisión sería satisfacer la demanda del cliente correspondiente y luego pasar a la otra locación. Sin embargo, estos documentos no proporcionan un resultado computacional.

Bianchi, Leonora, et al. en el **2006**[52] introduce una estrategia metaheurística de aproximación que mejora los resultados computacionales de Yang et al. en el 2000, teniendo una exigencia computacional bastante compleja. Recientes

---

[49] SHEN, Zhihong; ORDÓÑEZ, Fernando y DESSOUKY, Maged M. The Stochastic Vehicle Routing Problem for Minimum Unmet Demand. En: Optimization and Logistics Challenges in the Enterprise Springer Optimization and Its Applications. 2009. vol, 30.p 350.
[50] NOVOA. y STORER, Op. cit., p 510.
[51] ERERA, Alan L; MORALES, Juan C y SAVELSBERGH, Martin. The Vehicle Routing Problem with Stochastic Demand and Duration Constraints. En: The Logistics Institute Georgia Institute of Technology School of Industrial and Systems Engineering Atlanta [en línea]. (mar. 2009). [Consultado 3 Jun. 2013]. Disponible en <http://www2.isye.gatech.edu/~mwps/publications/VRPSD_DC_02Mar09.pdf>
[52] GOODSON, THOMAS y OHLMANN, Op. cit., p 5.

trabajos realizados consideran la aplicación de cinco metaheurísticas base para un solo vehículo: búsqueda local, búsqueda tabú, recocido simulado, colonia de hormigas y algoritmos evolutivos, dando mejores resultados que la heurística cíclica.

Siendo este un problema importante tanto operativo como estratégico y muy difícil de resolver en tamaños realistas, **Campbell** en el **2006**[53] genera una nueva alternativa para problemas de ruteo de vehículos con demandas estocásticas con tiempo limitado, *agregando* la mejor forma de dividir a los clientes en regiones de entrega y la escala necesaria para representar el objeto completo, así mismo el problema es reducido en su complejidad. La atención se centra en los grandes casos de problemas donde los clientes tienen una baja probabilidad de requerir una visita y el tiempo computacional disponible es bastante limitado. Con agregación se refiere a una agrupación de un conjunto de clientes, representados por la distribución de probabilidad y por un punto espacialmente, la ventaja clara de este método es que se da una reducción del tamaño del problema a priori.

K.C. Tan, C.Y. Cheong, y **C.K. Goh,**[54]en el **2007** estudia un algoritmo evolutivo multiobjetivo *(Multiobjetive Evolutionary Algorithm MOEA)* aplicado para resolver problemas VRPSD de optimización multiobjetivo, en contraste con los enfoques existentes de agregación, este algoritmo utiliza el concepto óptimo de Pareto para resolver esta clase de problemas, dando soluciones de mejora y la exploración de una búsqueda más grande. MOEA implementa dos heurísticas basadas en dos estructuras de ruta para la solución de problemas de ruteo de vehículo con demandas estocásticas para el método de búsqueda local: Shortest Path Search (SPS) y Which Directional Search (WDS). SPS, busca la ruta más corta aprovechando que la probabilidad de los fracasos en la ruta están más propensas a ocurrir al final de está, así mismo se intenta cambiar el orden de los clientes y

[53] CAMPBELL, Ann Melissa. Aggregation for the probabilistic traveling salesman problem. En: Computers and Operations Research,(2006);p 1-22.
[54] TAN, K.C; CHEONG, C.Y y GOH C.K.. En: European Journal of Operational Research. Vol. 177, (2007); p. 813-839.

WDS, teniendo en cuenta que el costo de transporte esperado depende de la trayectoria de la ruta e independientemente de la dirección que se elija, este será el mismo; el objetivo será construir una ruta nueva en dirección opuesta a la ruta ya planteada, sí ésta es de menor costo que la original entonces se reemplazará.

Ak, A. y **A. Erera** en el **2007**[55] presentan una estrategia de recurso dos a dos para el VRPSD, es decir, una estrategia en la que la flota de vehículos son coordinados en pares denotado *Paired Locally-Coordinated (PLC)*. El PLC es una estrategia de recurso que durante la trayectoria, cualquier suceso que indique una decisión de recurso puede resultar en una nueva planeación para los dos vehículos simultáneamente. Los resultados de esta estrategia es que cada cliente será servido por uno de los dos vehículos (el vehículo inicial o su socio), siendo el esfuerzo computacional no mucho mayor que la que se requiere para cuando el vehículo tiene que volver al depósito a reabastecerse. Según estudios realizados, se revela que la estrategia de recorrer la trayectoria en parejas puede dar lugar a ahorros en los costos de viaje esperados del 3% al 25% sobre los problemas VRPSD con 50 o más clientes.

Young, Peng y **Ha-Ying, Zhu**[56] en el **2008** utilizan un algoritmo DP (Programación Dinamica)- Optimización Enjambre de Partículas *(Particle Swarm Optimization PSO)* con operador Inver-over para dar solución a problemas VRPSD. El PSO es una herramienta de optimización basado en un proceso iterativo para buscar valores óptimos, en cada iteración se hace seguimiento de los dos valores "mejores", uno de ellos hará parte del individuo mejor y el otro valor se encontrara en la población y se denomina mejor mundial.

---

[55] Ak, A y A. Erera. A paired-vehicle routing recourse strategy for the vehicle routing problem with stochastic demands. En: Transportation Science. Vol. 41, No. 2, (2007); p 222–237.

[56] YOUNG, Peng y HA-YING, Zhu. Research on Vehicle Routing Problem with Stochastic Demandand PSO-DP Algorithm with Inver-over Operator. En: Systems Engineering- Theory & Practice. Vol. 10, (2008); p. 76-81.

5.4.3.1 Métodos de solución para el VRPSD basados en el RA. En el 2013[57] se presenta un marco de políticas de despliegue (RP) generalizado con el propósito de obtener resultados heurísticos para situaciones con demandas estocásticas. Unifican planteamientos de solución y motivados por la parte computacional formalizan variantes de despliegue que exploran los estados antes y las variables de decisión después.

El algoritmo de despliegue (RA) es una política de iteración heurística para problemas dinámicos que emplean el concepto *hacia delante* de programación dinámica, donde las decisiones se toman solo para los estados calculados observados. Desde el estado actual, la recompensa de ir se aproxima mediante las políticas heurísticas y esas aproximaciones se utilizan para guiar dicho estado.

El primer autor que proporciona una heurística computacional tratable **Secomandi** en **1998** y **Secomandi** en **2001**[58] investiga un enfoque de reoptimización del VRPSD, en el que se utiliza el RA para programación dinámica y así mejorar secuencialmente un recorrido a priori determinado, "en el que después de que es conocida la demanda de cada cliente, la parte restante del problema se vuelve a resolver por medio del uso de algoritmos de programación neuro-dinámica"[59]. Este enfoque genera mejores resultados que la estrategia de volver al depósito antes de que ocurra una acción que realmente lo obligue (estrategia preventiva de reposición de existencias) pero resulta ser mucho más costoso computacionalmente.

En el **2000**, **Secomandi**[60] estudia la aplicabilidad de la metodología de la programación neuro-dinamica *(Neuro Programation Dinamic, NPD)* a la formulación SSPP del VRPSD de un solo vehículo con el fin de calcular una política de enrutamiento dinámico óptimo. El desarrollo del problema compara dos

---

[57] GOODSON, THOMAS y OHLMANN, Op. cit., p 6.
[58] BIANCHI, Leonora., et al, Op.cit., p 4..
[59] TATARAKIS, A y MINIS, I. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. En: European Journal of Operational Research. Vol. 34, (2009); p. 558.
[60] SECOMANDI, Op cit., p 1201-1225.

algoritmos NPD: iteración política aproximada optimista *(Optimistic Approximate Policy Iteration, OAPI)* en el que una metodología de redes neuronales se utilizan para calcular las aproximaciones; y el algoritmo de despliegue de un solo paso *(One-step Rollout Algoithm ORA)* en el que la relación del costo de ir (Cost-to-go) se aproxima por medio de una heurística, los resultados computacionales muestran que la segunda de estas dos políticas es consistentemente y sustancialmente superior a la primera.

Secomandi en el **2003**[61] considera un problema de secuenciación que estudia un enfoque Rollout empleando una heurística cíclica como el algoritmo de base principal, garantizando resultados mejores y de calidad en la solución factible. Se tiene el problema de secuenciación en un VRPSD y se está interesado en la búsqueda de recorrido a través de un conjunto de clientes, se tiene una política dada en función de costo, el objetivo es encontrar la secuencia que minimiza dicha función con respecto a un conjunto de restricciones acerca de la secuencia buscada. Estudian teórica y computacionalmente una política de implementación de la secuenciación del problema basado en una heurística cíclica para generar, de esta forma, secuencias cuyos costos pueden ser interpretados como aproximaciones óptimos del costo de ir *(cost-to-go)* con relación a la calidad y el precio.

Novoa, Clara y **Storer, Robert**[62] en el **2009** asumen un enfoque de solución dinámica que modela los problemas en múltiples etapas llamado: reoptimización. En este enfoque las entregas se dividen cuando una ruta falla, es decir, el vehículo toma la decisión de completar la demanda del cliente en otra visita que no necesariamente tiene que ser inmediata. Se asume un algoritmo de despliegue *(Rollout Algorithm RA)* como un método heurístico eficiente para solucionar problemas del VRPSD en tiempo real. El RA es un método aproximado para

[61] SECOMANDI, Nicola. Analysis of a Rollout Approach to Sequencing Problems with Stochastic Routing Applications. En: Journal of heuristics. (2003); p. 321-352.
[62] NOVOA. y STORER, Op. cit., p 511.

resolver grandes problemas de PD que supone iteraciones para mejorar una política subóptima π sobre los estados del sistema.

6. PROGRAMACIÓN DINÁMICA

La programación dinámica, introducida por Bellman (1957), fue creada inicialmente para resolver problemas formulados en tiempo discreto, aunque posteriormente sería adaptada para la resolución de problemas en tiempo continuo[63]. La programación dinámica es una técnica matemática utilizada para desarrollar problemas en procesos de decisión de múltiples pasos, es decir, problemas donde se tomen decisiones de manera secuencial con el objetivo de minimizar el costo total de dichas decisiones. De manera que, el problema se descompone en sub-problemas, la solución de estos es almacenada para ser utilizada posteriormente y a partir del principio de optimalidad de Bellman ("todo subconjunto de una solución óptima es a su vez una solución óptima de un problema parcial") se construye un algoritmo de solución del problema en cuestión.

Algunos ejemplos de problemas secuenciales en donde se puede hacer uso de la programación dinámica son[64]:

- Búsqueda del camino más corto entre dos puntos
- Planificación de tareas
- Gestión de recursos escasos
- Gestión de stocks
- Coordinación hidrotermal

[63] CERDÁ, Emilio. Optimización Dinámica. 1 ed. México: Alfaomega Grupo Editor, 2012. p. 215-217.
[64] RAMOS, Andres. Programación dinámica. [en línea]; p 25-26. [Consultado 23 Ene. 2014]. Disponible en: < http://www.iit.upcomillas.es/aramos/simio/transpa/t_dp_ar.pdf>.

6.1 CARACTERÍSTICAS DE LOS PROBLEMAS DE PROGRAMACIÓN DINÁMICA[65]

1) Naturaleza secuencial de las decisiones: El problema se puede dividir en **etapas** y se requiere una decisión en cada etapa, estas hacen referencia a la trayectoria que hay de un estado al otro, los cuales deben estar definidos.

2) Cada etapa requiere de una **política de decisión**: Esto es, la decisión que se debe tomar en cada etapa para convertir el estado actual en un estado asociado con el inicio de la siguiente etapa.

3) Cada etapa tiene cierto **número de estados** asociado con su inicio: Los estados son las diferentes condiciones en las que se puede encontrar el sistema en cada etapa del problema, estos pueden ser finitos o infinitos.

4) El procedimiento de solución está diseñado para hallar un conjunto de políticas de decisión óptimas para cada etapa en cada uno de los estados posible y así encontrar una **política óptima** para manejar el problema completo.

5) El **principio de optimalidad** de programación dinámica define que, dado el estado actual, una política óptima para las etapas restantes es independiente de la política adoptada en etapas anteriores. Por lo tanto, la decisión inmediata óptima depende solo del estado actual y no de cómo se llegó ahí.

6) **Ecuación de recurrencia o función recursiva** debe indicar como se acumula la función de beneficios a optimizar (función objetivo) y como varían las funciones de estado de una etapa a otra.

---
[65] HILLIER, Frederick y LIEBERMAN, Gerald. Introducción a la Investigación de Operaciones. 9 ed. México: Mc Graw Hill. p. 445-447.

6.2 PLANTEAMIENTO DEL PROBLEMA DE CONTROL ÓPTIMO EN TIEMPO DISCRETO[66]

Se considera un sistema dinámico, formulado en tiempo discreto, para un número dado *N* de etapas o períodos, cuya situación inicial viene dada por el vector *n*-dimensional x_0 y que evoluciona en el tiempo. Dicha evolución depende del valor que se dé a ciertas variables, llamadas variables de control, que permiten influir en el sistema.

- $\boldsymbol{N} \rightarrow Número\ de\ etapas$
- $\boldsymbol{k} \rightarrow Etapa\ actual, para\ \boldsymbol{k} \in \{0,1,2,\ldots,\boldsymbol{N-1}\}$
- $\boldsymbol{x}\ \boldsymbol{k} \rightarrow \text{vector } n - \text{Dimensional, vector de } \textbf{variables de estado}\ en\ la\ etapa\ \boldsymbol{k}$
- $\boldsymbol{u}\ \boldsymbol{k} \rightarrow vector\ m - Dimensional, vector\ de\ \boldsymbol{variables\ de\ control}\ en\ la\ etapa\ \boldsymbol{k}$
- $\boldsymbol{\Omega}\ \boldsymbol{k} \rightarrow Conjunto\ de\ controles\ admisibles$

La evolución del sistema en el tiempo viene descrita por un sistema de ecuaciones en diferencias finitas, conocidas como ecuaciones de estado:

$$\boldsymbol{x}\ \boldsymbol{k+1} = \boldsymbol{f}\ \boldsymbol{x}\ \boldsymbol{k}, \boldsymbol{u}\ \boldsymbol{k}, \boldsymbol{k}, \boldsymbol{para}\ \boldsymbol{k} = \boldsymbol{0}, \boldsymbol{1}, \ldots, \boldsymbol{N-1}$$

$$con : x\ 0 = x_0,$$

Donde f es una función:

$$\boldsymbol{f}: \boldsymbol{D_1} \subset \mathbb{R}^{\boldsymbol{n}}\ \boldsymbol{x}\ \boldsymbol{D_2} \subset \mathbb{R}^{\boldsymbol{m}}\ \boldsymbol{x}\ \ \boldsymbol{0}, \ldots, \boldsymbol{N-1} \rightarrow \mathbb{R}^{\boldsymbol{n}}$$

$$\boldsymbol{x}, \qquad \boldsymbol{u}, \qquad \boldsymbol{k} \rightarrow \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{k})$$

Se supone que

---

[66] CERDÁ, Emilio. Op.cit., p 215

$$u(k) \in \Omega(k) \subset \mathbb{R}^m, para\; k = 0,1,\dots,N-1,$$

en donde para cada $\boldsymbol{k}$, $\boldsymbol{\Omega}(\boldsymbol{k})$ es el conjunto de controles admisibles.

La función objetivo es del tipo:

$$\boldsymbol{J} = \sum_{\boldsymbol{k}=\boldsymbol{0}}^{\boldsymbol{N}-\boldsymbol{1}} \boldsymbol{F}(\boldsymbol{x}(\boldsymbol{k}), \boldsymbol{u}(\boldsymbol{k}), \boldsymbol{k}) + \boldsymbol{S}(\boldsymbol{x}(\boldsymbol{N})),$$

siendo $F\; y\; S$ funciones definidas como:

$$\boldsymbol{f}: \boldsymbol{D_1} \subset \mathbb{R}^{\boldsymbol{n}}\; \boldsymbol{x}\; \boldsymbol{D_2} \subset \mathbb{R}^{\boldsymbol{m}}\; \boldsymbol{x}\; \{\boldsymbol{0}, \dots, \boldsymbol{N}-\boldsymbol{1}\} \rightarrow \mathbb{R}$$

$$(\boldsymbol{x}, \qquad \boldsymbol{u}, \qquad \boldsymbol{k}) \rightarrow \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{k})$$

y

$$\boldsymbol{S}: \boldsymbol{D_1} \subset \mathbb{R}^{\boldsymbol{n}} \rightarrow \mathbb{R}$$

$$\boldsymbol{x} \rightarrow \boldsymbol{S}(\boldsymbol{x}),$$

respectivamente.

Para determinar la evolución del sistema dinámico, se parte del estado inicial $\boldsymbol{x}_0$. En el período o etapa 1 (correspondiente a $\boldsymbol{k} = \boldsymbol{0}$), hay que elegir un control $\boldsymbol{u}(\boldsymbol{0}) \in \boldsymbol{\Omega}(\boldsymbol{0})$, en dicha etapa se realiza un aporte a la función objetivo dada por $\boldsymbol{F}[\boldsymbol{x}(\boldsymbol{0}), \boldsymbol{u}(\boldsymbol{0}), \boldsymbol{0}]$ y se inicia un período o etapa 2 (correspondiente a $\boldsymbol{k} = \boldsymbol{1}$) con el siguiente valor del vector estado.

$$\boldsymbol{x}(\boldsymbol{1}) = \boldsymbol{f}(\boldsymbol{x}(\boldsymbol{0}), \boldsymbol{u}(\boldsymbol{0}), \boldsymbol{0})$$

En dicho período o etapa 2 (correspondiente a $\boldsymbol{k} = \boldsymbol{1}$), hay que elegir un control $\boldsymbol{u}(\boldsymbol{1}) \in \boldsymbol{\Omega}(\boldsymbol{1})$, se realiza un aporte a la función objetivo dada por $\boldsymbol{F}[\boldsymbol{x}(\boldsymbol{1}), \boldsymbol{u}(\boldsymbol{1}), \boldsymbol{1}]$ y se inicia el período o etapa 3 (correspondiente a $\boldsymbol{k} = \boldsymbol{2}$) con el siguiente valor del vector de estado:

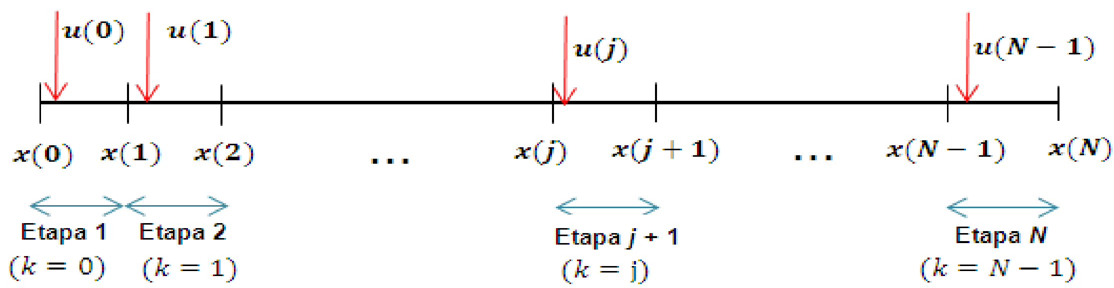
$$x(2) = f(x(1), u(1), 1)$$

Se sigue de esta manera hasta que por último comienza la etapa *N* (correspondiente a $k = N - 1$) con estado inicial $x(N-1)$, hay que elegir un control $u(N-1) \in \Omega(N-1)$, se realiza un aporte a la función objetivo dada por $F[x(N-1), u(N-1), N-1]$, alcanzando el sistema un estado final

$$x(N) = f(x(N-1), u(N-1), N-1),$$

y, por el hecho de terminar en dicho estado, se realiza un aporte a la función objetivo dada por $S[x(N)]$. Vease figura 9.

Figura 9. Control óptimo en tiempo discreto.



Fuente: Adaptado a partir de CERDÁ, Emilio. Optimización Dinámica. 1 ed. México: Alfaomega Grupo Editor, 2012. p. 217.

Un control óptimo se define como un control

$$u(k) \in \Omega(k) \subset \mathbb{R}^m, para\ k = 0,1, \ldots, N-1$$

(Admisible), que maximiza o minimiza la función objetivo.

Un problema de programación dinámica se puede formular como:

Dado un sistema dinámico con condición inicial $\boldsymbol{x_0}$*, y que evoluciona en el tiempo de acuerdo con la ecuación de estado* $\boldsymbol{x(k+1) = f(x(k), u(k), k)}$*, se trata de encontrar para cada* $\boldsymbol{k}$ *un vector de control que sea admisible y haga que la función objetivo alcance el valor máximo o minímo según sea el caso. Expresado en términos matemáticos se tratará de:*

$$\boldsymbol{max_{\{u(k)\}_{k=0}^{N-1}} J = \sum_{k=0}^{N-1} F(x(k), u(k), k) + S(x(N))},$$

Sujeto a: $x(k+1) = f(x(k), u(k), k)$, para $\boldsymbol{k = 0, 1, \ldots, N-1,}$

con: $\boldsymbol{x(0) = x_0}$ y $\boldsymbol{u(k) \in \Omega(k).}$

La secuencia de controles $\boldsymbol{u^* = (u^*(0), u^*(1), \ldots, u^*(N-1))'}$ que resuelve el problema se llama **control óptimo** y $\boldsymbol{x^* = (x^*(0), x^*(1), \ldots, x^*(N-1), x^*(N))'}$, determinado por la ecuación de estado a partir de $\boldsymbol{u^*}$ y de $\boldsymbol{x(0)}$, se llama **trayectoria de estado óptima** o camino óptimo.

6.3 PPROBLEMA DEL AGENTE VIAJERO (TSP)

El problema del agente viajero (*Traveling Salesman Problem, TSP*) es un problema clásico de optimización combinatoria. Las aplicaciones de la investigación de operaciones (IO) se encuentran en todo tipo de industrias y utilizar adecuadamente las técnicas de IO da soporte al complejo proceso de toma de decisiones que enfrentan las empresas, teniendo un impacto económico significativo. El TSP pertenece a la clase de problemas muy difíciles de resolver (NP), por lo tanto, es común usar métodos heurísticos para determinar soluciones factibles de alta calidad (relativamente cercanas al óptimo) en tiempos de ejecución relativamente pequeños.

Ejemplo:

Es el último fin de semana de la campaña de elecciones y el candidato Juan Acevedo está en Bogotá. Antes del día de elecciones, Juan debe visitar, Bucaramanga, Cartagena y Medellín, y luego regresar a sus oficinas centrales en Bogotá. Juan desea minimizar la distancia total que debe recorrer. ¿En qué orden debe visitar las ciudades?

Sabemos que Juan tiene que visitar cada ciudad exactamente una vez, que la última ciudad que visite debe ser Bogotá y que su viaje empieza en Bogotá.

La etapa se señalara por el número de ciudades que Juan ya visitó. En cualquier etapa se necesita saber dos cosas para determinar qué ciudad debe ser la siguiente en ser visitada; la ubicación actual de Juan y las ciudades que ya visitó. En cualquier etapa, el estado consiste en la última ciudad visitada y en el conjunto de ciudades que ya fueron visitadas. A continuación se presentan algunos supuestos:

- $\boldsymbol{t}$ Número de etapas, ciudades que Juan ya visitó.
- $\boldsymbol{i}$ Última ciudad visitada en el estado actual.
- $\boldsymbol{S}$ Conjunto de ciudades que ya fueron visitadas, $\boldsymbol{S = 1, 2, \ldots, t - 1}$
- $\boldsymbol{f_t(i, S)}$ Función de estado.
- $\boldsymbol{d_{ij}}$ Distancia entre las dos ciudades $\boldsymbol{i}$ y $\boldsymbol{j}$

Tabla 1. Distancias que debe recorrer el agente viajero

DISTANCIA ENTRE CIUDADES EN KM				
	Bogotá	Medellín	Cartagena	Bucaramanga
Bogotá	0	444	1052	407
Medellín	444	0	639	398
Cartagena	1052	639	0	651
Bucaramanga	407	398	651	0

Solución:

En este caso, el número de ciudades a visitar son **3**, por lo que se tendrán $\mathbf{3}! = \mathbf{6}$ soluciones factibles posibles.

<u>Etapa 1</u>

En la primera etapa Juan se encuentra en la ciudad **1**, por lo tanto tendrá que dirigirse a cualquiera de las tres ciudades restantes.

$$\boldsymbol{f_1}\ \mathbf{1},\ \cdot \quad = \quad \begin{array}{c} \boldsymbol{d_{12}} = \mathbf{444} \\ \boldsymbol{d_{13}} = \mathbf{1052} \\ \boldsymbol{d_{14}} = \mathbf{407} \end{array}$$

Etapa 2

Ahora, Juan tomo la decisión de ir a la ciudad **2** o a la ciudad **3** o a la **4**. En caso de que actualmente se encuentre en la ciudad **2**, él tendrá que dirigirse hacía alguna de las dos *ciudades restantes*, sea **3** o **4.** De igual manera se analiza, dependiendo de la ciudad actual.

- $$f_2\ 2,\ \ 2 \quad = min \begin{array}{l} d_{23} + f_1\ \ 1,\ \ \cdot \\ d_{24} + f_1\ \ 1,\ \ \cdot \end{array} \quad = \begin{array}{c} 639 + 444 \\ 398 + 444 \end{array} = \begin{array}{c} 1083 \\ 842 \end{array}$$

- $$f_2\ 3,\ \ 3 \quad = min \begin{array}{l} d_{32} + f_1\ \ 1,\ \ \cdot \\ d_{34} + f_1\ \ 1,\ \ \cdot \end{array} \quad = \begin{array}{c} 639 + 1052 \\ 651 + 1052 \end{array} = \begin{array}{c} 1691 \\ 1703 \end{array}$$

- $$f_2\ 4,\ \ 4 \quad = min \begin{array}{l} d_{42} + f_1\ \ 1,\ \ \cdot \\ d_{43} + f_1\ \ 1,\ \ \cdot \end{array} \quad = \begin{array}{c} 398 + 407 \\ 651 + 407 \end{array} = \begin{array}{c} 805 \\ 1058 \end{array}$$

Etapa 3

En la etapa 3, Juan ya ha pasado por dos ciudades, sin contar la ciudad de origen y le queda por recorrer una ciudad antes de llegar nuevamente al origen. Sí la ciudad actual en esta etapa es la **2**, él pudo haber llegado desde la ciudad **3** o **4**, de igual forma se analiza para las demás.

- $$\begin{array}{l} f_3\ 2,\ \ 3,2 \quad = \min\ d_{24} + f_2\ \ 3,\ \ 3 \quad = \quad 398 + 1691 \ \ = 2089 \\ f_3\ 2,\ \ 4,2 \quad = \min\ d_{23} + f_2\ \ 4,\ \ 4 \quad = \quad 639 + 805 \ \ = 1444 \end{array}$$

- $$\begin{array}{l} f_3\ 3,\ \ 2,3 \quad = \min\ d_{34} + f_2\ \ 2,\ \ 2 \quad = \quad 651 + 1083 \ \ = 1734 \\ f_3\ 3,\ \ 4,3 \quad = \min\ d_{32} + f_2\ \ 4,\ \ 4 \quad = \quad 639 + 1058 \ \ = 1697 \end{array}$$

- $$\begin{array}{l} f_3\ 4,\ \ 2,4 \quad = \min\ d_{43} + f_2\ \ 2,\ \ 2 \quad = \quad 651 + 842 \ \ = 1493 \\ f_3\ 4,\ \ 3,4 \quad = \min\ d_{42} + f_2\ \ 3,\ \ 3 \quad = \quad 398 + 1703 \ \ = 2101 \end{array}$$

Etapa 4

En la última etapa Juan puede estar en cualquiera de las tres ciudades para luego pasar al origen y estando en la ciudad actual, este pudo haber llegado desde cualquiera de las dos ciudades restantes.

- $f_4\ 2,\ 3,4,2 \quad = min \begin{matrix} d_{21} + f_3\ 4,\ 3,4 \\ d_{21} + f_3\ 3,\ 4,3 \end{matrix} \quad = \quad \begin{matrix} 444 + 2101 \\ 444 + 1697 \end{matrix} \quad = \quad \begin{matrix} 2545 \\ 2141 \end{matrix}$

- $f_4\ 3,\ 2,4,3 \quad = min \begin{matrix} d_{31} + f_3\ 4,\ 2,4 \\ d_{31} + f_3\ 2,\ 4,2 \end{matrix} \quad = \quad \begin{matrix} 1052 + 1493 \\ 1052 + 1444 \end{matrix} \quad = \quad \begin{matrix} 2545 \\ 2496 \end{matrix}$

- $f_4\ 4,\ 2,3,4 \quad = min \begin{matrix} d_{41} + f_3\ 3,\ 2,3 \\ d_{41} + f_3\ 2,\ 3,2 \end{matrix} \quad = \quad \begin{matrix} 407 + 1734 \\ 407 + 2089 \end{matrix} \quad = \quad \begin{matrix} 2141 \\ 2496 \end{matrix}$

Existen dos rutas en las cuales se minimiza la distancia recorrida, estas son:

Ruta 1 = 1-4-3-2-1; Ruta 2 = 1-2-3-4-1

6.4 MODELO DE PROGRAMACIÓN DINÁMICA ENFOCADO AL ALGORITMO ROLLOUT

Los problemas de optimización se pueden resolver usando la Programación Dinámica, naturalmente su uso más difundido es cuando hay problemas de optimización con decisiones secuenciales en etapas sucesivas donde el objetivo es minimizar el costo total de dichas decisiones.

En este caso, el árbol o modelo de programación dinámica es necesario para la compresión de la resolución del algoritmo Rollout y sus distintas variables. Más que mirar un resultado final, el objetivo del modelo es dar a conocer los posibles escenarios y la complejidad que se presenta en la resolución del algoritmo, observando las trayectorias resultantes, controles, etapas, estados del sistema que se dan en este proceso de solución del problema.

6.4.1 Planteamiento del problema. La siguiente notación es tenida en cuenta, en un contexto general, para la solución del problema a tratar. Se define[67]:

- $\boldsymbol{N} \rightarrow Número\ de\ estapas, variable\ aleatoria$
- $\boldsymbol{S} \rightarrow Espacio\ de\ estado.$
- $\boldsymbol{k} \rightarrow Etapa\ actual, para\ \boldsymbol{k} \in \{0,1,2,\dots,\boldsymbol{N-1}\}$
- $\boldsymbol{Q} \rightarrow Capacidad\ del\ vehículo$
- $\boldsymbol{u_k} \rightarrow Representa\ el\ control\ o\ desición\ tomada\ en\ la\ etapa\ k, dado\ por\ \boldsymbol{u_k}\ \boldsymbol{m}, \boldsymbol{a}$
  - $\boldsymbol{m} \in\ 1,\dots,n\ |r_m \neq 0\ \cup\ 0\ , es\ cualquier\ cliente\ próximo\ a\ visitar$
  - $\boldsymbol{a} \in\ 0{,}1\ ;$

$$a\ \begin{matrix} \boldsymbol{0}\ El\ vehículo\ visita\ al\ cliente\ directamente \\ \boldsymbol{1}\ El\ vehículo\ para\ primero\ en\ el\ depósito\ y\ luego\ va\ al\ elcliente \end{matrix}$$

- $\boldsymbol{g}\ \boldsymbol{x_k}, \boldsymbol{u_k} \rightarrow Costos\ de\ transición\ en\ la\ etapa\ k, donde:$
  - $d\ l, m \rightarrow\ Costo\ directo, yendo\ directamente\ al\ cliente$
  - $d\ l, 0\ + d\ 0, m \rightarrow Costo\ indirecto, yendo\ primero\ al\ depósito$ $y\ luego\ al\ cliente$
- $\boldsymbol{x_k} \rightarrow Estado\ del\ sistema\ en\ la\ estapa\ k, dado\ por\ x_k = (l, q_l, r_1, \dots, r_n)$
  - $l\ \rightarrow La\ localización\ actual\ del\ vehículo, donde\ l\ \in 0, 1, \dots n$
  - $q_l\ \rightarrow\ Capacidad\ residual\ del\ vehículo\ en\ el\ cliente\ l,$ $donde\ q_l \in 0, 1, \dots, Q$
  - $r_i \rightarrow Representa\ la\ demanda\ aún\ no\ entregada\ al\ cliente\ i$

En esta clase de problemas las demandas solo se conocen cuando el vehículo llega a la locación del cliente, entonces, la demanda aún no conocida del cliente en el estado del sistema se representa por ?, y toma valores entre $1, 2, \dots, R$, y es cero para el cliente que se visita y que ha sido servido en forma completa, esto es, la demanda se satisface totalmente. Así, el estado inicial del sistema se representa como $(0, Q, ?, ?, ?)$ y el estado final del sistema será $(0, Q, 0{,}0{,}0)$. El

[67] NOVOA. y STORER, Op. cit., p 510.

recorrido del vehículo empieza y termina en el depósito, teniendo en cuenta que durante la trayectoria se presentarán fallas de rutas y retornos proactivos.

A continuación se plantea un ejemplo el cual mostrará los diferentes escenarios que se generan en la solución del problema de ruteo de vehículo empleando la programación dinámica.

Se tiene un vehículo con capacidad $Q = 5$, el cual visitará a tres posibles clientes $(n = 3)$, los nodos representan el conjunto de clientes $1, 2, \dots n$ y el depósito es representado por el nodo cero. Sabiendo que los clientes se encuentran geográficamente dispersos, se presenta las distancias entre ellos en la tabla 2 y cada cliente tiene dos demandas posibles las cuales se ilustran en la tabla 3:

Tabla 2. Matriz de distancias

	Depósito	**Cliente 1**	**Cliente 2**	**Cliente 3**
Depósito	0	1334	1559	809
Cliente 1	1334	0	1343	1397
Cliente 2	1559	1343	0	921
Cliente 3	809	1397	921	0

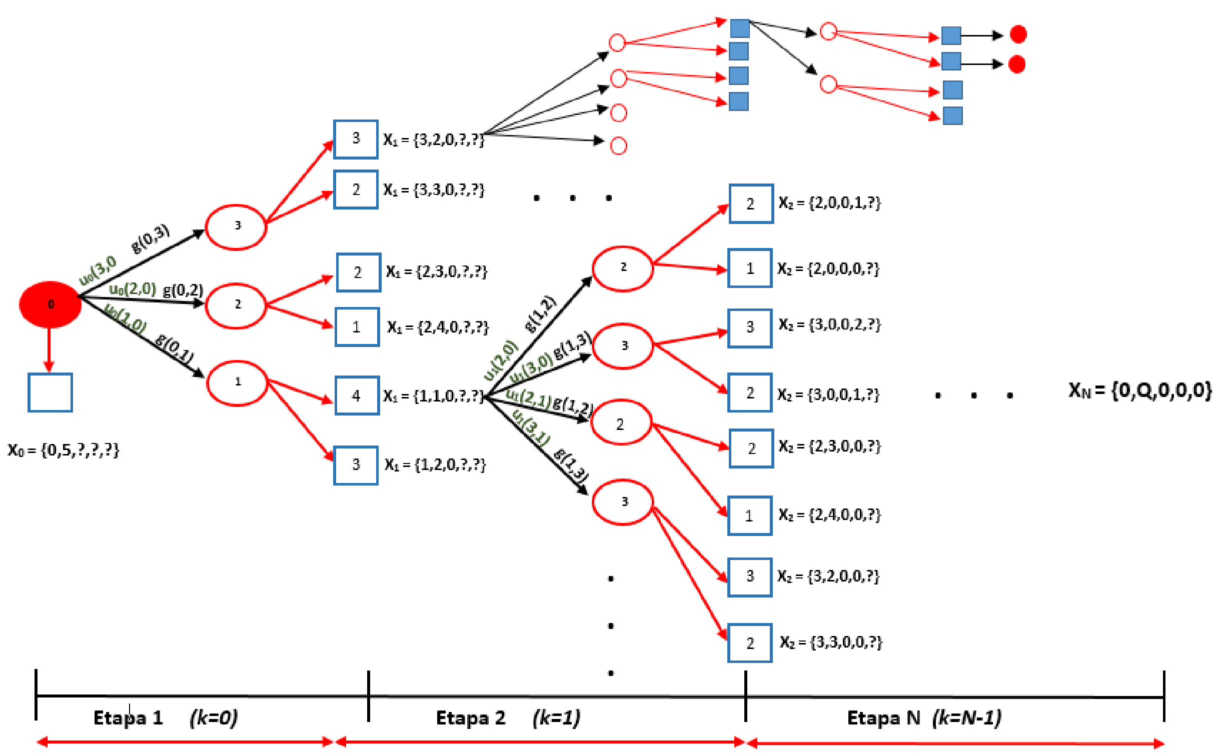
Tabla 3. Demanda de los clientes

Clientes	Demandas	
1	4	3
2	2	1
3	3	2

Donde $D_i, i = 1,2,\ldots,n,$, es la demanda del cliente y es una variable aleatoria.

En la siguiente figura se muestra el árbol de programación dinámica.

Figura 10. Árbol de programación dinámica



En esta ilustración el recorrido comienza en el depósito con un estado del sistema inicial $x_0 = \{0,5,?,?,?\}$, donde la locación del vehículo es el depósito, el vehículo se encuentra con una capacidad total y las tres demandas de los clientes a visitar son aún desconocidas. La primera etapa, donde $k = 0$, el vehículo se puede dirigir a cualquiera de los tres clientes posibles. En el trascurso de esta etapa ocurren controles o decisiones definidos como $u_0 \ m, 0$, donde m es el cliente próximo a visitar, que en este caso puede ser el cliente $1, 2$ o 3 y cero es el trayecto que hace el vehículo desde el depósito a los clientes directamente, ya que va con la capacidad total del vehículo. También, un costo de transición $g \ x_0, u_0$ se genera al estar en un estado x_0 y haber tomado el control u_0, el cual está relacionado con la distancia recorrida desde el depósito al cliente. Cuando el vehículo llega al cliente (sea $1, 2$ o 3), la demanda se revela y esta se suple, por lo tanto, el estado del sistema se actualiza. Es decir, si el vehículo llega al cliente 1 el estado del sistema sería $x_1 = (1, 1, 0, ?, ?)$ si la demanda del cliente es 4 y $x_1 = (1, 2, 0, ?, ?)$ si la demanda del cliente es 3, empezando así una nueva etapa.

La segunda etapa, donde $k = 1$, comienza cuando el estado del sistema se actualiza, si el vehículo se encuentra en el cliente 1, los clientes próximos a visitar serían 2 o 3. El vehículo debe dirigirse a los clientes tomando un control respectivo, en el caso de dirigirse hacía 2 el control sería $u_1 \ 2{,}0$ si va directamente al cliente o $u_1 \ 2{,}1$ si primero pasa por el depósito y luego se dirige al cliente 2. Si se supone que la demande del cliente 1 es 3, con un estado del sistema $x_1 = (1, 2, 0, ?, ?)$ y partimos de allí hacía el cliente 2, se observan dos estados del sistema, $x_2 = (2, 0, 0, 0, ?)$ si la demanda es 2 y $x_2 = (2, 1, 0, 0, ?)$ si la demanda es 3, empezando así una nueva etapa. Debido a los controles seleccionados en cada etapa, un costo $g \ x_k, u_k$ .se genera en cada una de ellas.

Este procedimiento se realiza para cada etapa teniendo en cuenta los estados del sistema que se generan debido a las demandas de cada cliente, hasta terminar el recorrido. Cada estado puede generar un escenario diferente, ya sea que el vehículo se quede sin existencias y deba retornar al depósito para poder seguir su ruta o que regrese al depósito para poder suplir la demanda faltante de un cliente.

6.5 PROGRAMACIÓN DINÁMICA PROBABILÍSTICA

Su principal característica es, que el estado de la siguiente etapa no está determinado por el estado y la política de decisión de la etapa actual. Por el contrario, existe una distribución de probabilidad conocida sujeta por el estado y la política de decisión de la etapa actual, que determina cuál será el siguiente estado[68].

En los problemas de programación dinámica, el conocimiento del estado actual del sistema expresa toda la información sobre el comportamiento anterior, la cual es necesaria para determinar la política óptima. Esta propiedad es llamada *propiedad markoviana.*

6.5.1 Descripción del problema[69]. Consideramos el siguiente sistema dinámico formulado en tiempo discreto:

$$x(k+1) = f(x(k), u(k), v(k), k), \qquad para\ k = 0, 1, 2, \ldots, N-1$$

$$Con: x(0) = x_0,$$

Donde para cada k:

- $\boldsymbol{x(k)} \rightarrow Vector\ de\ variables\ de\ estado, pertenecientes\ al\ espacio\ \boldsymbol{S}_k$
- $\boldsymbol{u(k)} \rightarrow Vector\ de\ varibles\ de\ control, perteneciente\ al\ espacio\ \boldsymbol{C}_k$

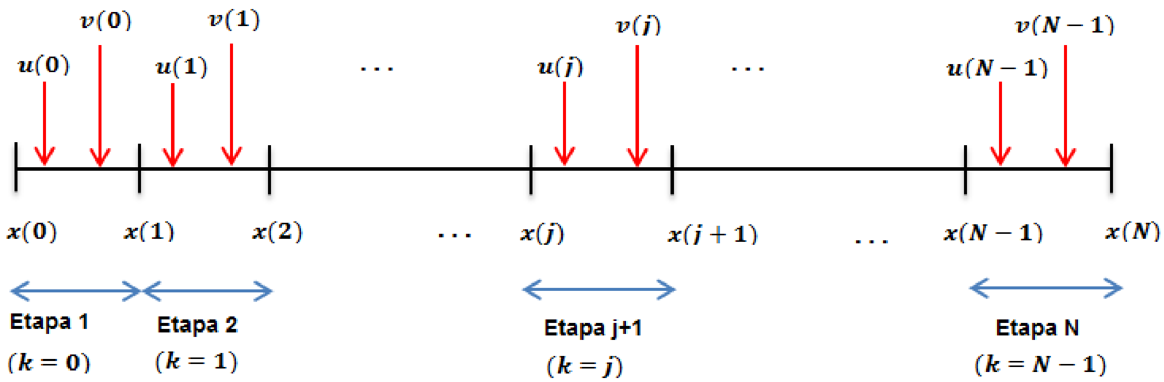
---

[68] HILLIER, Frederick y LIEBERMAN, Gerald. Op. cit., p 466.
[69] CERDÁ,Emilio, Op.cit., p 216.

- $\boldsymbol{v}\ \boldsymbol{k} \rightarrow Vector\ de\ perturbaciones\ aleatorias, perteneciente\ al\ espacio\ \boldsymbol{D}_k$

En la figura 11 se puede observar la secuencia en la que van a ir tomando valores las diferentes variables, a través del tiempo.

Figura 11. Etapas y variables en el problema de PDP



Fuente: Adaptado a partir de CERDÁ, Emilio. Optimización Dinámica. 1 ed. México: Alfaomega Grupo Editor, 2012. p. 206

El control $u(k)$ está restringido a tomar valores pertenecientes a un conjunto no vacío $\Omega_k[x(k)]$, que depende del estado del sistema en ese período $\boldsymbol{k}$, es decir:

$$u(k) \in \Omega_k[x(k)], para\ cada\ x(k) \in S_k, \forall k = 0, 1, \dots, N-1.$$

La perturbación aleatoria $v(k)$ viene caracterizada por una distribución de probabilidad $P_k(\bullet\ |x(k), u(k))$, que puede depender explícitamente de $x(k), u(k)$, pero es independiente de $v(k-1), v(k-2), \dots, v(0)$.

Consideremos la clase de leyes de control (también llamadas políticas), que consisten en una secuencia de funciones:

$$\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\},$$

en donde cada función μ_k transforma el estado $x(k)$ en el control $u(k) = \mu_k[x(k)]$, de manera que se verifica que

$$\mu_k[x(k)] \in \Omega_k[x(k)], para\ todo\ \ x(k) \in S_k.$$

Las leyes de control que cumplen esta condición se llamarán admisibles

El problema consiste, exactamente, en lo siguiente:

Dado un estado inicial x_0, se trata de encontrar una ley de control admisible

$$\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$$

que maximice el funcional siguiente:

$$\max J_\pi(x_0) = E\left\{\sum_{k=0}^{N-1} F[x(k), \mu_k[x(k)], v(k), k] + S[x(N)]\right\}$$

sujeto a: $x(k+1) = f[x(k), \mu_k[x(k)], v(k), k].\ para\ k = 0,1, \dots, N-1,$

con: $x(0) = x_0,$

$\mu_k[x(k)] \in \Omega_k[x(k)], para\ todo\ \ x(k) \in S_k$,

en donde E{•} significa esperanza matemática. Por otra parte, las funciones $F, S\ y\ f$ están dadas.

Entonces, se trata de encontrar una ley de control óptimo π^*, para la cual se verifique:

$$J_{\pi^*}(x_0) = \max_{\pi \in \Pi} J_\pi(x_0),$$

donde Π es el conjunto de leyes de control admisibles.

Entonces la función J^*, definida de la siguiente forma,

$$J^*(x_0) = J_{\pi^*}(x_0),$$

es la función que asigna a cada estado inicial x_0 el valor objetico óptimo $J^*(x_0)$, y se llama **función valor óptimo**.

7. ALGORITMO ROLLOUT

El VRPSD se ha estudiado bajo dos enfoques distintos, **Bertsimas et al**. en **1990** presenta un problema que es tratado de manera *estática*, mientras que **Dror et al.** en **1989** y **Dror** en **1993** maneja un tipo de problema *dinámico* donde el conjunto de clientes está sujeto a cambios durante el servicio. El algoritmo rollout es el conjunto de pasos lógicos para realizar una constante optimización de problemas de programación dinámica complejos y encontrar una política óptima de manera iterativa, donde las decisiones se toman solo para los estados calculados observados. Inicialmente el Rollout fue propuesto por **Bertsekas et al**. en **1997** y **Bertsekas** en **2001** y **2000** e implementado por **Secomandi** en **1998, 2000** y **2001** para resolver problemas dinámicos del VRPSD[70].

El rollout es considerado un algoritmo de programación neuro-dinamica (*Neuro-dynamic Programming NPD*). La NPD, también conocida como aprendizaje de refuerzo, es una metodología que trata de solucionar problemas grandes y complejos de programación dinámica, en donde combina la simulación, el aprendizaje, las arquitecturas de aproximación (por ejemplo, redes neuronales) y las ideas centrales de la programación dinámica para romper la maldición de la dimensionalidad[71].

Dentro de la NPD se encuentran dos algoritmos basados en la iteración de política aproximada (*Approximate Policy Iteration API)* que son: API optimista (*optimistic Approximate Policy Iteration OAPI*) y el algoritmo rollout de un solo paso (*one-step rollout algorithm ORA*). OAPI aproxima la función óptima del costo de ir para todos los estados del sistema como funciones lineales o no lineales de problemas con características pre-seleccionadas (locación del vehículo, capacidad del vehículo, etc.). La idea principal es generar un número pequeño de trayectorias bajo una

[70] NOVOA. y STORER, Op. cit., p 510.
[71] SECOMANDI, Op.cit., p 1203-1204

política inicial resolviendo problemas con mínimos cuadrados. Mientras que el ORA consiste en obtener una mejor política a partir de la política de base, aplicando un cambio de un solo paso a la vez que se va hacia adelante[72].En este caso se aplicará el ORA.

7.1 DESCRIPCIÓN DEL PROBLEMA

Existe un vehículo con capacidad fija Q, este parte del depósito para dirigirse hacia sus clientes, los cuales se encuentran ubicados en lugares distantes, entregando o recogiendo un bien. Entones, se denota que:

- $\boldsymbol{I} \rightarrow Representa\ el\ conjunto\ de\ clientes, donde\ I = 1, 2, \dots, n$
- $\boldsymbol{Nodo}\ \mathbf{0} \rightarrow Hace\ referencia\ al\ depósito$
- $\boldsymbol{d}\ \boldsymbol{i}, \boldsymbol{j} \rightarrow\ Distancia\ entre\ clientes$
- $\boldsymbol{D_i} \rightarrow Variable\ aleatoria\ que\ describe\ la\ demanda\ del\ cliente, donde$ $i = 1,2, \dots, n$
- $\boldsymbol{p_i} \rightarrow Distribución\ de\ probabilidad\ en\ función\ de\ la\ demanda, siendo$ $\boldsymbol{p_i}\ \boldsymbol{r} = P_r\ D_i = r\ , r = 0,1, \dots, R \leq Q$

Los costos calculados son proporcionales a las distancias recorridas en la ruta. La demanda de los clientes solo se conoce cuando el vehículo llega a su ubicación, como consecuencia, se pueden presentar *fallas de ruta* donde la demanda excede la capacidad del vehículo $D_i > Q$, por lo que el vehículo puede satisfacer parcialmente al cliente, ir al depósito para reabastecerse y seguir con la visita. Existe una política de ruteo donde el vehículo realiza una *ruta proactiva* en el cual se dirige al depósito a reabastecerse para recuperar su capacidad Q antes de que

---

[72] BERTSEKAS, Dmitri P. Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC. En: European Journal of Operational Research, (2005); p. 5.

este quede sin existencias y ocurra un fallo de ruta. El objetivo de dar solución al problema es, encontrar una política de enrutamiento que minimice los costos y satisfacer completamente la demanda de todos los clientes.

7.2 FORMULACIÓN DEL PROBLEMA[73]

Sea $X_k = \ l, q_l, r_1, \ldots, r_n$ una matriz con $\mathrm{n}+2$ componentes que representa el estado del sistema en la etapa de decisión $k,$ donde:

- $\boldsymbol{l} \rightarrow Ubicación\ actual\ del\ vehículo, l \in 0,1., \ldots, n.$
- $\boldsymbol{q_l} \rightarrow Capacidad\ actual\ del\ vehículo, q_l \ \in 0,1, \ldots, Q$
- $\boldsymbol{r_i} \rightarrow Demanda\ aún\ no\ entregada\ al\ cliente\ i, i = 1, 2, \ldots n$
    - $r_i =?, si\ la\ demanda\ no\ se\ conoce\ y\ puede\ tomar\ un\ valor\ cualquiera$ $0, 1, \ldots, R$
    - $r_i = 0, si\ el\ cliente\ ha\ sido\ visitado\ y\ la\ demanda\ se\ satisfe.$
    - $r_i' = 1, 2, \ldots, R-1, valor\ actual\ de\ la\ demanda\ de\ un\ cliente\ cuando$

    $este\ ha\ sido\ visitado\ y\ parcialmente\ servido.$

    $Al\ tomar\ un\ valor\ conocido\ p_r\ \ D_i = r' \ \ = 1$

El conjunto de estados (espacio) se denota como S. El número total de estados posibles en el sistema es elevado y es de orden $O\ \ nQR^n$. El sistema comienza en el estado $(0, Q, ?, ?, \ldots, ?)$, el vehículo se encuentra en el nodo cero (depósito), con una capacidad total y demandas desconocidas. El estado final $0, Q, 0,0, \ldots, 0$, el vehículo vuelve al depósito recuperando la capacidad total y ha suplido las demandas de los clientes totalmente.

La variable aleatoria N representará el número de etapas desde el estado inicial hasta el estado final, su distribución de probabilidad se determina a partir de la

[73] NOVOA. y STORER, Op. cit., p 511.

distribución de las probabilidades de las demandas de los clientes por la ruta seguida o asignada. El sistema pasa a un nuevo estado cuando el vehículo llega (no necesariamente por primera vez) a una ubicación del cliente.

La matriz $\pi = \mu_o, \mu_{1,}, \ldots, \mu_{N-1}$ es la política o secuencia de funciones a optimizar donde μ_k es una función asociada al control o decisión $u_k = \mu_k(x_k)$ para cada estado. Un control u_k hace parte del conjunto de controles o decisiones factibles para el estado x_k , este indica la decisión que se toma en cada etapa durante el recorrido. De tal manera que:

$u_k \in U_k(x_k)$ y $U_k(x_k) = \quad m \in \; 1, 2, \ldots, n \quad r_m \neq 0 \; \cup 0\} \times \{a\colon a \in \{0, 1\}\}$.

- $\boldsymbol{u_k} \rightarrow Representa\ el\ control\ tomado\ en\ la\ etapa\ k, dado\ por\ \boldsymbol{u_k}\ \boldsymbol{m}, \boldsymbol{a}$
  - $\boldsymbol{m} \in \; 1, \ldots, n \; |r_m \neq 0 \; \cup \; 0$, $es\ cualquier\ cliente\ próximo\ a\ visitar.$

m toma el valor de 0 cuando las demandas son satisfechas y el sistema entra en el estado final.

  - $\boldsymbol{a} \in \; 0{,}1$, es la decisión de ir al depósito o no.

a toma el valor de 0 si el vehículo visita al cliente directamente y toma el valor de 1 si el vehículo se devuelve al depósito para luego continuar con el cliente.

En el caso de $a = 1$ se permiten viajes proactivos a la estación evitando fallas en las rutas, con el fin de reducir los costos.

Si el sistema está en la etapa k y en el estado $X_k = \; l, q_l, r_1, \ldots, r_n \; \in S$ y se elige el control u_k entonces, el sistema pasa al estado $x_{k+1} = \; m, q_m, r_1, \ldots, r'_m, \ldots, r_n$, con costo de transición $g \; x_k, u_k$ y probabilidad de transición $p \; x_k, x_{k+1} \; (u_k)$. El costo de transición $g(x_k, u_k)$ es proporcional a la distancia entre dos clientes, bien sea de ir en forma directa o indirecta (esto es, primero al depósito y luego regresando al cliente).

La capacidad del vehículo después de servir al cliente m, es igual a q_m, ahora la demanda r'_m, del cliente m, se actualiza de acuerdo con la realización de la demanda del cliente. El objetivo del problema es encontrar una política π que minimice el costo de ir J_N^{π} (*cost-to-go)* en la $N-etapa,$ desde algún estado inicial. El costo óptimo de ir en la $N-etapa$ (es decir, costo esperado de terminación) para el estado x es $J_N^*\ x\ = min_{\pi\in}\ \ J_N^{\pi}(x)$ donde es el conjunto de políticas admisibles.

Si el valor J_N^* es conocido para todos los estados, el control óptimo u_k^* para cada estado, resulta de la búsqueda del mínimo en la siguiente ecuación:

$$u_k^*\ x\ = \mu_k^*\ x\ = arg \min_{u_k \in U_k\ x_k} g\ x_k, u_k\ + \sum_{x_{k+1}\in S} p_{x_k x_{k+1}}\ u_k\ J_N^*\ x_{k+1}\ |$$

$$x_k = x, \forall x \in S \quad (1)$$

La ecuación (1) muestra que en todas las etapas, las decisiones son clasificadas con base a la suma del precio actual y el precio futuro esperado, asumiendo una toma de decisión óptima para las etapas subsiguientes. A partir de esta sección se simplificará el subíndice N en la relación del costo de ir para simplificar la notación.

En el algoritmo rollout de una sola etapa (ORA), se construye un control aproximado u_k para un estado particular x_k en la etapa k, con el objetivo de aproximar el costo optimo $J^*\ x_{k+1}$ (de ir) en la ecuación (1) por $J(x_{k+1})$. Esta aproximación se realiza para todos los estados $x_{k+1} \in S$ que se pueden alcanzar en un paso hacia delante de la etapa actual k[74]. Debido a lo difícil de alcanzar el valor óptimo del problema, el algoritmo rollout es una buena alternativa de hallar una solución aproximada.

Denotando como $J(x_k)$ el costo de ir siguiendo la política de base en el estado x_k; se elige un control de acuerdo con la siguiente minimización:

[74] NOVOA. y STORER, Op. cit., p 511.

$$u(x_k) = \arg\min_{u \in U(x_k)} \left[g(x_k, u_k) + \sum_{x_{k+1} \in S} P_{x_k x_{k+1}}(u_k) \cdot J(x_{k+1}) \right] \qquad (2)$$

Si se dispone de una evaluación exacta de $J(x_{k+1})$, entonces se demuestra que la política de despliegue mejora la calidad de la política común (Bertsekas y Tsitsiklis 1996). Es importante señalar que la aplicación eficaz de este método requiere que $J(x_{k+1})$ sea calculada eficientemente[75].

7.3 LA POLÍTICA ROLLOUT PARA EL VRPSD

La complejidad en la solución del VRP se debe a la cantidad de rutas posibles que dependen del número de clientes, por lo tanto, al aumentar el número de clientes el problema crece en complejidad, y por consiguiente, representa un mayor uso de recursos como el tiempo y espacio (cantidad de memoria). El número de rutas posibles es igual a $n!$, donde *n* es el número de clientes que se deben servir. Cuando se analiza el VRPSD, el problema no solo radica en todas las posibles rutas, ahora se adiciona el hecho de que los clientes poseen demandas estocásticas y esto hace que incremente la dificultad del problema. En la literatura científica se encuentra una gran cantidad de métodos de solución, y entre ellos se encuentra el algoritmo rollout que es el tema de investigación del presente trabajo. El algoritmo Rollout representa una de las respuestas a la búsqueda exhaustiva de procedimientos para la solución del VRPSD de una manera aproximada y la optimización de recursos.

[75] SECOMANDI, Nicola. A rollout policy for the vehicle routing problem with stochastic demands. En: Institute for Operation Research and de Management Science. Vol. 49, No. 5, (2001); p 798.

El Rollout se aplica siempre y cuando exista una secuencia de base o tour base τ para el problema y calcula el costo de ir esperado $J(x_{k+1})$ eficientemente. La selección del tour a priori de manera correcta, hace que el Rollout arroje una política de mejora τ[76]. Para la aplicación de este algoritmo, se emplea una heurística cíclica ζ eficiente y consistente, Bertsekas y Tsitsiklis en 1996 y Bertsekas et al.(1973) afirman que si la ruta base es calculado con la heurística cíclica, la política resultante es mejor que la política inicial.

7.3.1 Secuencia de base o política a priori. El primer paso para la implementación del Rollout es encontrar la secuencia de base por medio de la heurística Nearest Neighbor (NN) para el TSP, definida como:

Dado un conjunto de puntos $P = \ p_1, \dots, p_n$ en un espacio vectorial X, se da un nuevo punto de consulta $q \in X$, encontrar en el conjunto P el punto que este más cercano a q eficientemente[77]. La ruta del vehículo comienza y termina en el depósito (denotado por el nodo 0); la ruta construida por medio del NN se describe a continuación: para el conjunto de clientes $I = 1,2, \dots, n$, se encuentra el cliente más próximo al depósito, ignorando las demandas de los clientes y aplicando el vecino más cercano. Una vez elegido el cliente más cercano al depósito, a partir de allí se debe repetir el procedimiento para los clientes faltantes hasta completar una ruta llamada secuencia de base (SB), tour base o política a priori denotada por τ.

Se considera la secuencia de base $\tau = (0, 1, 2, \dots n, 0)$ donde el vehículo visita a los clientes siguiendo está ruta, se tiene en cuenta los retornos al depósito en caso de un fallo de ruta o al darse reposiciones tempranas durante el recorrido. Al seguir

[76] NOVOA. y STORER, Op. cit., p 512.
[77] MARIUS, Muja y LOWE David G. Fast approximate nearest neighbors with automatic algorithm configuration. En: Computer Science Department, University of British Columbia, Vancouver, B.C., Canada; p 1.

esta ruta, se realizan actualizaciones de la secuencia de base que ayudan a recalcular el costo aproximado de ir.

7.3.2 Cálculo de los controles en la política Rollout. El cálculo de los controles se realizan, a lo largo del recorrido, en lo que respecta a la elección del siguiente cliente y la manera de ir hasta él.

A partir de la ruta a-priori obtenida por el Nearest Neighbor (NN); las visitas del vehículo a los clientes se realizan en el orden dado, y la selección se hace de acuerdo a la demanda del cliente actual. En caso de que se seleccione el siguiente cliente se decide ir al cliente o bien volver al depósito para un reabastecimiento.

Suponga que el servicio en el cliente actual j ha terminado y el vehículo tiene una carga q restante, se define la función $f_j(q)$ como el costo total esperado del nodo j en un viaje hacia adelante. El costo esperado de la ruta a-priori es igual a $f_0(Q)$. Si L_j representa el conjunto de todas las cargas posibles que un vehículo puede tener después de la terminación del servicio en el cliente j, entonces, $f_j\ q$ para $q \in L_j$ satisface[78]:

$$\boldsymbol{f_j\ q\ = \min\ f_j^p\ q\ ,\ f_j^r\ q} \qquad \textbf{(3)}$$

donde

$$f_j^p\ q\ = d_{j,j+1} + \quad {}_{k:k\leq q} f_{j+1}(q-k)p_{j+1,k} + \quad {}_{k:k>q}\ 2d_{j+1,0} + f_{j+1}(q+Q-k)\ p_{j+1,k} \quad \textbf{(4)}$$

$$f_j^r\ q\ = d_{j,0} + d_{0,j+1} + \quad {}_{k=1}^{K} f_{j+1}(Q-k)p_{j+1,k} \quad \textbf{(5)}$$

---

[78] BIANCHI, Leonora, et al. Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Journal of Mathematical Modelling and Algorithms. (2006); p. 91-110.

Con la condición límite:

$$f_n \; q \; = d_{n,0}, \; q \in L_n \qquad \textbf{(6)}$$

Siendo:

- $\boldsymbol{f_j^p \; q}$ $\rightarrow$ Costo esperado de ir directamente al cliente
- $\boldsymbol{f_j^r \; q}$ $\rightarrow$ Costo esperado de ir al cliente haciendo primero un retorno al depósito
- $\boldsymbol{d_{ij}} \rightarrow Distancia\ entre\ nodos$
- $\boldsymbol{p_{j+1,k}} \rightarrow Probabilidad\ de\ que\ la\ demanda\ en\ el\ nodo\ \boldsymbol{j+1}\ sea\ igual\ a\ \boldsymbol{k}$
- $\boldsymbol{k} \rightarrow Demanda\ del\ cliente$
- $\boldsymbol{q} \rightarrow Capacidad\ residual\ del\ vehículo$

Se debe tener en cuenta que en la fórmula (4) si $k \leq q$ se aplica $d_{j,j+1} + {}_{k:k\leq q} f_{j+1}(q-k)p_{j+1,k}$, y/o si $k > q$ se aplica $d_{j,j+1} + {}_{k:k>q}\ 2d_{j+1,0} + f_{j+1}(q+Q-k)\ p_{j+1,k}$. La fórmula (6) es el resultado de ir desde el último cliente por visitar hasta el depósito, valor que será remplazado de atrás hacia adelante en cada función indicada f_{j+1} de cada etapa.

7.4 DESARROLLO DEL ALGORITMO ROLLOUT

Se tiene una secuencia de base $\tau = (0, 1, 2, \ldots n, 0)$ y se asume que el vehículo visita a los clientes de acuerdo con τ, retornando al depósito en caso en que hayan fallas durante la ruta, o si se realiza un reabastecimiento preventivo. Bertsimas (1992) propone una heurística cíclica ζ que realiza actualizaciones en la secuencia de base cambiando el orden de dicha secuencia (en un solo paso ORA) y construyendo una ruta cíclica o secuencia de base actualizada (*Updated Base*

Sequence UBS) τ_t^{ζ} que comienza en la locación t y sigue a τ cíclicamente, omitiendo los clientes que se encuentran completamente servidos[79].

$$\tau_t^{\zeta} = (t, t+1, \dots, n, 1, \dots, t-1, 0)$$

La distancia esperada o el costo de ir aproximado de τ_t^{ζ} en el estado x_k es $J(x_{k+1})$, el cual hace parte de la ecuación (2) descrita en la formulación del problema, este costo se asocia a la distancia y se halla usando las ecuaciones (4), (5) y (6).

Partiendo del depósito el vehículo se dirige al primer cliente, siguiendo la secuencia de base a priori τ, tomando siempre un control de $u_k(m, 0)$ para la primera etapa, la demanda del cliente t es observada y completamente servida (porque $k \leq Q$)[80], luego la capacidad del vehículo se actualiza. El sistema se mueve a un estado x_{k+1} y se calcula el control para seguir el recorrido y decidir hacia qué cliente ir. En cada estado x_k el vehículo debe tomar un control u_k en donde después de haber servido al cliente actual t, se origina una secuencia de base actualizada τ_t^{ζ}.

Se determina luego la distancia esperada para cada secuencia de base actualizada, se procede a escoger el de menor valor entre $\{f_j^p\ q\ ,\ f_j^r\ q\ \}$ y este valor es comparado junto con las otras secuencias base actualizadas, donde el mínimo en esa etapa es el control que se debe seleccionar para pasar al estado x_{k+1} (siguiendo τ). El procedimiento se realiza a lo largo de la ruta a-priori y sobre la secuencia base actualizada óptima, hasta que todos los clientes han sido totalmente servidos, el vehículo retorna al depósito y resulta una ruta final mejorada τ junto con un conjunto de controles admisibles mejorados π.

La distancia total esperada de la ruta final, teniendo en cuenta cada control u_k, se halla de la siguiente manera:

---

[79] SECOMANDI, Op.cit., p 798.
[80] Ibid.,p. 797.

- Si en el estado x_k un control $\boldsymbol{u_k}\ \boldsymbol{m}, \mathbf{0}$ se selecciona, entonces la distancia que se calcula en esta etapa será $\boldsymbol{d}(\boldsymbol{l}, \boldsymbol{m})$

- Si en el estado x_k un control $\boldsymbol{u_k}\ \boldsymbol{m}, \mathbf{1}$ se selecciona, entonces la distancia que se calcula en esta etapa será $\boldsymbol{d}\ \boldsymbol{l}, \mathbf{0}\ + \boldsymbol{d}(\mathbf{0}, \boldsymbol{m})$

8. RESULTADOS NUMÉRICOS

En el presente capítulo se discute los factores y niveles utilizados para el banco de pruebas, luego se generan las instancias que usan en el análisis de los resultados que se obtienen con la herramienta desarrollada en el presente proyecto. La herramienta hace referencia al programa elaborado en Micosoft Excel para la simulación del VRPSD, aplicando como método de solución el Algoritmo Rollout.

8.1 GENERACIÓN DE INSTANCIAS

Una instancia comprende la asignación de los valores a un conjunto de parámetros característicos del problema. Por medio de las instancias se analizan los resultados obtenidos por el método de solución propuesto.

Los parámetros para el VRPSD tenidos en cuenta en la generación de las instancias son:

- Capacidad del vehículo (Q)
- Número de clientes
- Demanda promedio de cada cliente
- Probabilidad de las demanda
- Ubicación de los clientes

En el caso de problemas clásicos como el VRP, existen diversas fuentes de instancias o conjuntos de datos en la literatura, sin embargo, Bianchi *et al.*[81] señalan que comúnmente no se utiliza el Benchmark para el VRPSD. Por tanto, varios autores han desarrollado sus propios bancos de pruebas, por ejemplo

[81] BIANCHI, Leonora, et al. Op. cit. p. 98.

"Yang *et al*. y Teodorovic y Lucic, generaron aleatoriamente la ubicación del depósito y de los clientes, la demanda de cada cliente, la media y la varianza."[82]

Utilizando la propuesta de investigación realizada por Silvia Galván[83] se ha construido un banco de pruebas, adaptando los factores y niveles al presente problema para validar el Algoritmo Rollout.

Para la generación de las instancias se determinan dos niveles para cada uno de los factores. El número de clientes toma los valores de $n = [4, 8, 12]$ para los niveles 1 y 2. La ubicación de los clientes en el nivel 1 presenta una distribución uniforme dentro del intervalo [0, 20] y en el nivel 2 se presenta una distribución normal. La desviación tomada, con respecto a la demanda, está comprendida en el intervalo de [1, 5] para el nivel 1 y [80, 100] para el nivel 2. Por tal razón, las cuatro posibles demandas de cada cliente toma valores en el intervalo de $[D_i - S_i, D_i + S_i]$ y la probabilidad de ocurrencia para cada una de las posibles demandas de los clientes presenta una distribución de probabilidad uniforme.

La capacidad del vehículo se calcula de acuerdo a la siguiente formula[84]:

$$Q = \frac{{}_{i=1}^{n}(D_i) * r}{n}$$

Donde:

- r es el número promedio de clientes atendidos por el vehículo antes de ir al depósito. Debido a que el número máximo de clientes a introducir en el programa es de 12, se realizó para cada grupo de clientes una prueba en la cual se varia el r para observar el valor más adecuado a utilizar dentro del problema Rollout, teniendo en cuenta que la capacidad del vehículo debe ser mayor a la demanda máxima que posean los clientes y que los controles

---

[82] TAN, K.C.; CHEONG, C.Y. y GOH, C.K. Op cit p.813–839
[83] GALVÁN, Silvia; ARIAS, Javier y LAMOS, Henry. En: DYNA. Junio, 2013. Ed,179. pp. 60-69.
[84] GENDREAU, M., LAPORTE, G. y SÉGUIN, R., An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers. En: Transportation Science.1995.pp. 143-155.

resultantes reflejen las idas al depósito, ya que con una capacidad muy grande para el problema el vehículo cumplirá la ruta sin necesidad de hacer retornos al depósito. A continuación en la tabla 4 se muestra la capacidad del vehículo para cada grupo de clientes:

Tabla 4. Capacidad del vehículo promedio para cada grupo de clientes variando $\boldsymbol{r}$

Número de clientes	**Número de clientes atendidos antes de ir al depósito**		
	$\boldsymbol{r = 2}$	$\boldsymbol{r = 4}$	$\boldsymbol{r = 8}$
4	117	235	470
8	121	241	482
12	121	241	482

De acuerdo a lo anterior, se escoge el valor de $r = 2$, ya que con este valor se observa una capacidad del vehículo conveniente para cada grupo de clientes y poder analizar la interacción que existe entre factores y niveles dentro del problema.

- D_i representa la demanda promedio para cada cliente i, el valor de r es el número promedio de clientes atendidos por el vehículo antes de ir al depósito. En este caso toma el valor de 2 los dos niveles. Se debe tener en cuenta que la capacidad del vehículo siempre debe ser mayor o igual a la máxima demanda de los clientes.

A continuación se presenta la tabla 5 de factores y niveles:

Tabla 5. Factores y niveles

	Nivel 1	Nivel 2
Número de clientes (n)	4, 8, 12	4, 8, 12
Ubicación Clientes (p)	[0, 20] Distribución uniforme	(10, 5) Distribución normal
Demanda promedio (D_i	[50, 70]	[80, 100]
Desviación (S_i)	[1, 5]	[10, 15]

El banco de pruebas está conformado por ocho instancias para cada conjunto de clientes $n = [4, 8, 12]$, entonces, en total existen veinticuatro instancias generadas y cada una contiene una combinación de factores y niveles. A continuación se presenta las tablas 6, 7 y 8 con la información de las instancias para cada conjunto de clientes.

Tabla 6. Banco de pruebas para 4 clientes

Instancias	Número de clientes	Ubicación de los clientes	Demanda promedio	Desviación
1	4	Nivel 1	Nivel 1	Nivel 1
2	4	Nivel 2	nivel 1	Nivel 1
3	4	Nivel 1	Nivel 2	Nivel 2
4	4	Nivel 2	Nivel 2	Nivel 2
5	4	Nivel 1	Nivel 1	Nivel 2
6	4	Nivel 1	Nivel 2	Nivel 1
7	4	Nivel 2	Nivel 1	Nivel 2
8	4	Nivel 2	Nivel 2	Nivel 1

Tabla 7. Banco de pruebas para 8 clientes

Instancias	Número de clientes	Ubicación de los clientes	Demanda promedio	Desviación
9	8	Nivel 1	Nivel 1	Nivel 1
10	8	Nivel 2	Nivel 1	Nivel 1
11	8	Nivel 1	Nivel 2	Nivel 2
12	8	Nivel 2	Nivel 2	Nivel 2
13	8	Nivel 1	Nivel 1	Nivel 2
14	8	Nivel 1	Nivel 2	Nivel 1
15	8	Nivel 2	Nivel 1	Nivel 2
16	8	Nivel 2	Nivel 2	Nivel 1

Tabla 8. Banco de pruebas para 12 clientes

Instancias	Número de clientes	Ubicación de los clientes	Demanda promedio	Desviación
17	12	Nivel 1	Nivel 1	Nivel 1
18	12	Nivel 2	nivel 1	Nivel 1
19	12	Nivel 1	Nivel 2	Nivel 2
20	12	Nivel 2	Nivel 2	Nivel 2
21	12	Nivel 1	Nivel 1	Nivel 2
22	12	Nivel 1	Nivel 2	Nivel 1
23	12	Nivel 2	Nivel 1	Nivel 2
24	12	Nivel 2	Nivel 2	Nivel 1

9. DISEÑO DE EXPERIMENTO

9.1 FACTORES DEL DISEÑO DE EXPERIMENTO

Para el análisis de los factores que influyen en la respuesta de la función objetivo, se eligió el diseño factorial 2^3, en donde cada factor se estudia a dos niveles y se contemplan todas las combinaciones de cada nivel de un factor con todos los niveles de los otros factores. En este caso se consideran tres factores: ubicación de los clientes, demanda promedio, desviación y dos niveles para cada uno de ellos: nivel 1 y nivel 2. En este caso el algoritmo no tiene parámetros definidos, por tanto, para el problema del Rollout se trabaja con tres factores del VRPSD. El número de réplicas para cada instancia es de 4. A continuación se muestra los 8 tratamientos que se realizarán para cada grupo de clientes.

Tabla 9. Tratamientos del diseño factorial 2^3

	A Ubicación	B Demanda	C Desviación
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

9.2 RESULTADOS DEL DISEÑO EXPERIMENTAL

El algoritmo Rollout fue programado en Excel 2010 y ejecutado en un equipo procesador Intel Core i5 con 6GB de RAM. A continuación se muestra los resultados obtenidos de las instancias al ser ejecutados en el software para validar la información obtenida. Cada instancia tiene 5 réplicas las cuales se han hecho para tomar un costo y tiempo promedio. Las tablas 10, 11 y 12 muestran los resultados del diseño de experimento para cada grupo de clientes.

Tabla 10. Resultado del diseño experimental para 4 clientes.

Instancias	Corridas			Valor promedio de la función objetivo para 4 clientes
	A	B	C	
Instancia 1	-	-	-	89,175526
Instancia 2	+	-	-	114,59144
Instancia 3	-	+	-	90,7667
Instancia 4	+	+	-	114,02512
Instancia 5	-	-	+	94,470784
Instancia 6	+	-	+	106,116574
Instancia 7	-	+	+	90,451154
Instancia 8	+	+	+	118,233002

Tabla 11. Resultado del diseño experimental para 8 clientes.

Instancias	Corridas			Valor promedio de la función objetivo para 8 clientes
	A	B	C	
Instancia 9	-	-	-	167,49716
Instancia 10	+	-	-	183,36254
Instancia 11	-	+	-	190,45842
Instancia 12	+	+	-	196,64172
Instancia 13	-	-	+	162,87122
Instancia 14	+	-	+	177,60888
Instancia 15	-	+	+	186,74602
Instancia 16	+	+	+	191,54096

Tabla 12. Resultado del diseño experimental para 12 clientes.

Instancias	Corridas			Valor promedio de la función objetivo para 12 clientes
	A	B	C	
Instancia 17	-	-	-	234,958
Instancia 18	+	-	-	241,38144
Instancia 19	-	+	-	250,07362
Instancia 20	+	+	-	267,00408
Instancia 21	-	-	+	249,63766
Instancia 22	+	-	+	243,59108
Instancia 23	-	+	+	246,904
Instancia 24	+	+	+	243,47142

9.3 ANÁLISIS DEL DISEÑO EXPERIMENTAL

9.3.1 Análisis de efectos estimados para cada grupo de clientes. A cada grupo de clientes del banco de pruebas se le aplicó un análisis de diseño de experimento seleccionado. A continuación se presenta los efectos estimados de cada uno de los factores para cada grupo sobre la variable respuesta.

Tabla 13. Efectos estimados para la variable respuesta para 4 clientes

Factores	Efecto estimado para 4 clientes
A: Ubicación clientes	23,8663
B: Demanda promedio	1,04194
C: Desviación	-4,95912

Tabla 14. Efectos estimados para la variable respuesta para 8 clientes

Factores	Efecto estimado para 8 clientes
A: Ubicación clientes	13,3646
B: Demanda promedio	26,5669
C: Desviación	-2,90337

Tabla 15. Efectos estimados para la variable respuesta para 12 clientes

Factores	Efecto estimado para 12 clientes
A: Ubicación clientes	0,26692
B: Demanda promedio	7,31942
C: Desviación	-4,5636

El signo positivo de un efecto estimado indica que al pasar de un nivel bajo a un nivel alto el costo de la ruta aumenta, esto se refleja en los factores A y B de cada grupo de clientes, mientras el factor C tiene un valor negativo. El objetivo del problema es minimizar el costo de la ruta recorrida, por tanto, la ubicación de los clientes y la demanda promedio deben optar un nivel bajo y la desviación tomar un nivel alto.

Los efectos de cada grupo de clientes se grafican en un diagrama de Pareto para observar los factores que tienen más impacto sobre la variable de respuesta.

Figura 12. Diagrama de Pareto para 4 clientes.

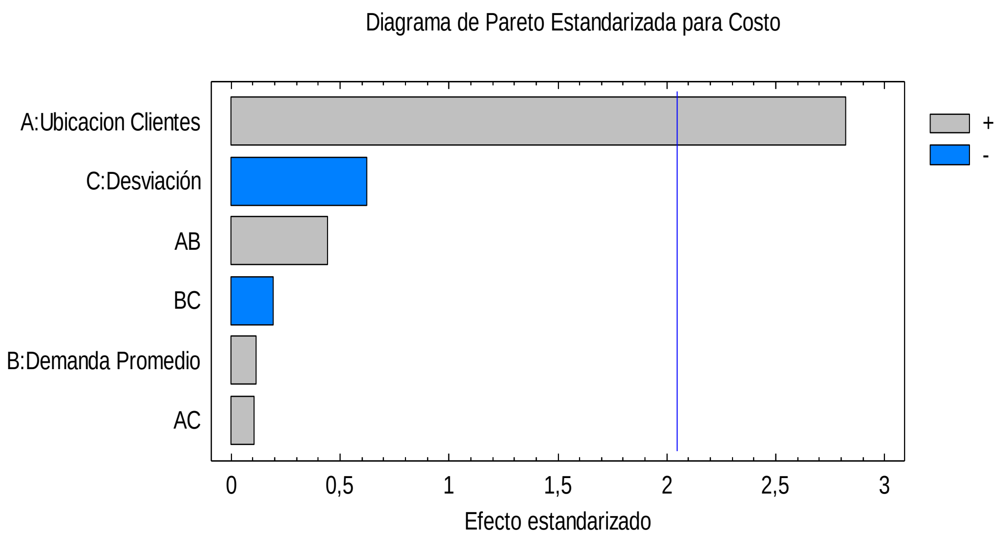


Figura 13. Diagrama de Pareto para 8 clientes.

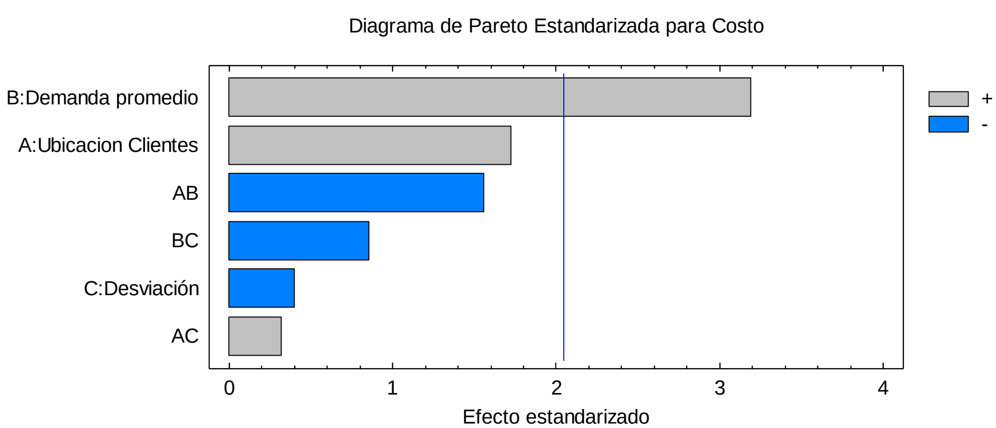
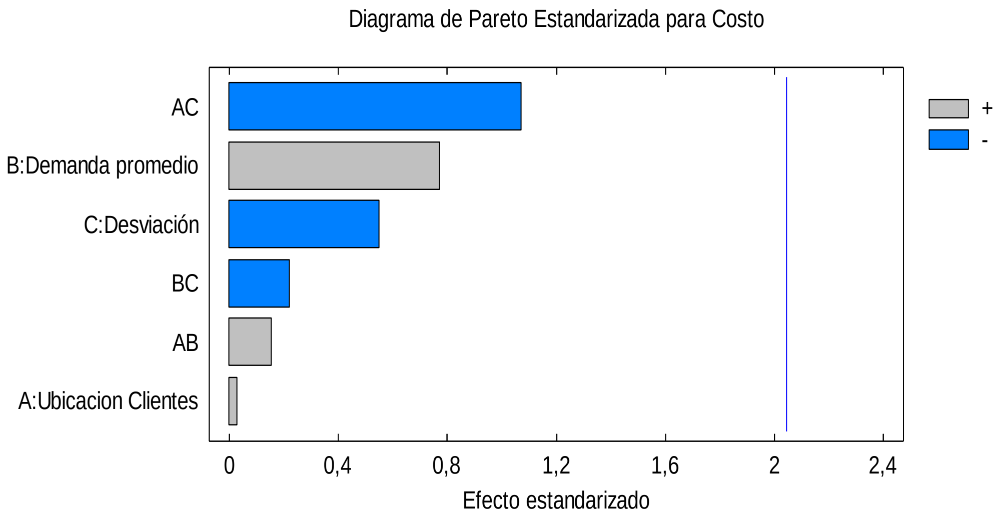


Figura 14. Diagrama de Pareto para 12 clientes



Se observa en el diagrama de Pareto la importancia de los efectos para cada grupo de clientes. La ubicación de los clientes y la demanda promedio es un efecto significativo para los grupos de clientes 4 y 8 respectivamente. Además se observa que la desviación tiene un efecto no significativo sobre la variable de respuesta para cada grupo de clientes, lo que indica que este puede tomar cualquiera de los dos niveles. Se puede observar que los efectos principales del problema no son de mayor incidencia sobre la función objetivo en el grupo de 12 clientes.

9.3.2 Análisis de varianza (ANOVA) para cada grupo de clientes.

En la Tabla 16 se observa que la ubicación de los clientes (A) es el único factor que arroja un Valor-P menor a 0.05, por tanto el efecto generado es significativo sobre la función objetivo. Cuando la ubicación de los clientes usa un nivel bajo la función objetivo tiende a encontrar la mejor solución. Los demás factores y las interacciones son de poca incidencia sobre la variable respuesta.

Tabla 16. Análisis de varianza para 4 clientes.

Fuente	Suma de Cuadrados	Grados de libertad	Cuadrado Medio	Razón-F	Valor-P (Significancia)
A:Ubicacion Clientes	1924,74	1	1924,74	7,96	0,0085
B:Demanda Promedio	3,17748	1	3,17748	0,01	0,9095
C:Desviación	94,4549	1	94,4549	0,39	0,5368
AB	47,6333	1	47,6333	0,2	0,6604
AC	2,70136	1	2,70136	0,01	0,9165
BC	9,05831	1	9,05831	0,04	0,8479
Error total	7009,7	29	241,714		
Total (corregido)	12368,4	39			

En la Tabla 17 se observa que el efecto de la demanda promedio (B) es significativo en la respuesta ya que el Valor-P es menor a 0.05. Mientras que la ubicación de los clientes y la desviación no tienen ningún tipo de incidencia en la respuesta del problema, por lo que se espera que la interacción AC no contribuya. Para minimizar la variable de respuesta la demanda promedio debe usar un nivel bajo.

Tabla 17. Análisis de varianza para 8 clientes.

Fuente	Suma de Cuadrados	Grados de libertad	Cuadrado Medio	Razón-F	Valor-P (Significancia)
A:Ubicacion Clientes	603,556	1	603,556	2,97	0,0954
B:Demanda promedio	2065,76	1	2065,76	10,17	0,0034
C:Desviación	32,3757	1	32,3757	0,16	0,6927
AB	495,409	1	495,409	2,44	0,1292
AC	21,3612	1	21,3612	0,11	0,7481
BC	148,441	1	148,441	0,73	0,3996
Error total	5890,88	29	203,134		
Total (corregido)	11336,2	39			

De la Tabla 18 se puede concluir que tanto las interacciones como los factores no influyen sobre la variable de respuesta ya que el valor de significancia para todos está por encima de 0.05.

Tabla 18. Análisis de varianza para 12 clientes.

Fuente	Suma de Cuadrados	Grados de libertad	Cuadrado Medio	Razón-F	Valor-P (Significancia)
A:Ubicacion Clientes	0,24075	1	0,24075	0	0,976
B:Demanda promedio	156,802	1	156,802	0,6	0,4454
C:Desviación	79,9892	1	79,9892	0,31	0,5848
AB	6,49748	1	6,49748	0,02	0,8759
AC	301,898	1	301,898	1,15	0,2919
BC	13,1366	1	13,1366	0,05	0,8244
Error total	7595,85	29	261,926		
Total (corregido)	10725,8	39			

9.3.3 Análisis de varianza (ANOVA) para las 24 instancias. Para cada instancia hay un costo promedio obtenido de las cuatro replicas realizadas durante el procedimiento del problema. El costo promedio es utilizado para analizar los factores influyentes sobre la función objetivo. A continuación el ANOVA para las 24 instancias.

Tabla 19. Análisis de varianza para 12 instancias

Fuente	Suma de Cuadrados	Grados de libertad	Cuadrado Medio	Razón-F	Valor-P (Significancia)
A:Ubicación clientes	850,112	1	850,112	12,17	0,0082
B:Demanda promedio	578,71	1	578,71	8,29	0,0206
C:Desviación	29,4169	1	29,4169	0,42	0,5345
AB	1,9925	1	1,9925	0,03	0,8701
AC	20,3241	1	20,3241	0,29	0,6043

Fuente	Suma de Cuadrados	Grados de libertad	Cuadrado Medio	Razón-F	Valor-P (Significancia)
BC	21,3445	1	21,3445	0,31	0,5955
Error total	558,8	8	69,85		
Total (corregido)	27572,5	15			

En la tabla 19 se observa que la ubicación de los clientes y la demanda promedio son significativos sobre la función objetivo ya que tienen un Valor-P menor a 0.05. Se obtendrán mejores resultados para la variable de respuesta cuando el factor A y B estén en un nivel bajo. La desviación y las interacciones entre los factores no inciden en la variable de respuesta.

9.4 ANÁLISIS DE LOS CONTROLES PARA LA POLÍTICA ROLLOUT

Los controles son decisiones que el vehículo toma en cada etapa de su recorrido aplicando el algoritmo Rollout para optimizar dichas decisiones, por lo que el costo total del recorrido depende de los controles.

El programa utilizado para la solución del VRPSD por medio del algoritmo Rollout tiene las siguientes entradas:

- Número de clientes que el vehículo debe visitar a lo largo de la ruta para poder suplir su demanda totalmente, siendo doce el número máximo de clientes.
- La demanda promedio. Las posibles demandas de cada cliente, las cuales pueden ser hasta cuatro, varían dependiendo de la demanda promedio y la desviación. En este caso, se trabajó con cuatro posibles demandas para cada uno de los clientes.
- Cada posible demanda de cada cliente presenta una probabilidad uniforme de ocurrencia, por lo tanto, un problema puede presentar varias soluciones a

la hora de realizar las corridas en el programa, ya que este escoge aleatoriamente una demanda de las cuatro posibles demandas que presenta el cliente.

- Desviación para la demanda.
- Capacidad del vehículo.
- Coordenadas para la ubicación de los clientes.

A continuación se tiene como objetivo analizar los controles arrojados por el programa y su comportamiento a lo largo del recorrido, eligiendo una ruta resultante de cada grupo de clientes y observando las veces que el vehículo tiene que volver al depósito a reabastecerse para suplir completamente las demandas de los clientes. Para el análisis de cada grupo de clientes (4, 8 y 12) se toma como referencia el hecho de que los clientes demandarán el valor máximo de las posibles cuatro demandas para cada uno.

9.4.1 Análisis de controles para cuatro clientes

- Ruta final: 0 - 1 - 3 - 4 - 2 - 0
- Controles resultantes: $\mu_0\ 1,0\ ,\mu_1\ 3,0\ ,\mu_2\ 4,1\ ,\mu_3\ 2,1\ ,\mu_4\ 0,0$

Tabla 20. Posibles demandas para 4 clientes

Capacidad (Q) 117	**Demanda 1**		**Demanda 2**		**Demanda 3**		**Demanda 4**	
	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad
Cliente 1	51	0,25	52	0,25	53	0,25	54	0,25
Cliente 2	57	0,25	59	0,25	55	0,25	56	0,25
Cliente 3	58	0,25	56	0,25	57	0,25	63	0,25
Cliente 4	67	0,25	68	0,25	66	0,25	65	0,25

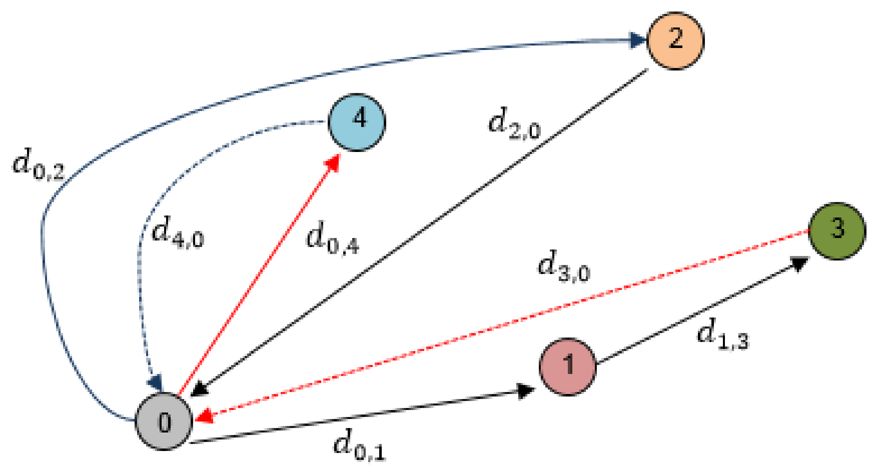
Teniendo en cuenta los controles resultantes de la aplicación del Rollout, se observa que el vehículo en la etapa cero se dirige directamente al cliente 1 cumpliendo la ruta a priori, siendo esta la base para comenzar el recorrido. Luego de suplir completamente al cliente 1 el vehículo se dirige a 3, esa decisión se toma con base a las distancias y las posibles demandas que tienen los clientes que aún faltan por visitar.

Como se muestra en la tabla 20, las demandas máximas para 4 clientes son 54, 59, 63 y 68 respectivamente, el vehículo después de atender a los clientes 1 y 3 queda con una capacidad residual de 4, por lo que debe retornar al depósito y realizar una recarga preventiva antes de dirigirse al siguiente cliente. Siguiendo el recorrido y luego de reabastecerse, el vehículo queda con una capacidad Q de 117 y se dirige al cliente 4, al servirlo la capacidad residual es de 49 y el único cliente que queda por visitar es el 2. Cuando se realizan los cálculos en la etapa 3, el vehículo decide ir al cliente 2 pero pasando antes por el depósito, ya que su capacidad residual es menor a lo que posiblemente demandaría este cliente. Después de haber servido al cliente 2, finalmente el vehículo retorna al depósito dejando satisfecho la totalidad de sus clientes.

El recorrido descrito anteriormente se ilustra en la figura 15, siendo esta una de las rutas resultantes que minimiza el costo del recorrido.

En la figura 15 se observa dos reabastecimientos preventivos, cuando el vehículo se dirige al cliente 4 y cuando se dirige al cliente 2. Esto quiere decir que al realizarse el cálculo se observa que la mejor decisión es ir al depósito antes de continuar la ruta, debido a que la capacidad residual puede que no supla completamente al cliente, esto con el propósito de que no suceda un fallo de ruta.

Figura 15. Ruta final con retornos para 4 clientes



9.4.2 Análisis de controles para ocho clientes

- Ruta final: 0 - 1 - 7 - 3 - 2 - 6 - 8 - 4 - 5 - 0
- Controles resultantes:

μ_0 $1,0$,μ_1 $7,0$,μ_2 $7,1$,μ_3 $3,0$,μ_4 $2,0$,μ_5 $2,1$,μ_6 $6,0$,μ_7 $8,0$,μ_8 $4,1$,μ_9 $5,1$,μ_{10} $0,0$

Tabla 21. Posibles demandas para 8 clientes

Capacidad (Q) 121	**Demanda 1**		**Demanda 2**		**Demanda 3**		**Demanda 4**	
	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad
Cliente 1	51	0,25	52	0,25	53	0,25	54	0,25
Cliente 2	57	0,25	59	0,25	55	0,25	56	0,25
Cliente 3	58	0,25	56	0,25	57	0,25	63	0,25

Capacidad (Q) 121	Demanda 1		Demanda 2		Demanda 3		Demanda 4	
	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad
Cliente 4	67	0,25	68	0,25	66	0,25	65	0,25
Cliente 5	63	0,25	64	0,25	65	0,25	66	0,25
Cliente 6	57	0,25	53	0,25	51	0,25	54	0,25
Cliente 7	69	0,25	71	0,25	68	0,25	67	0,25
Cliente 8	59	0,25	56	0,25	60	0,25	61	0,25

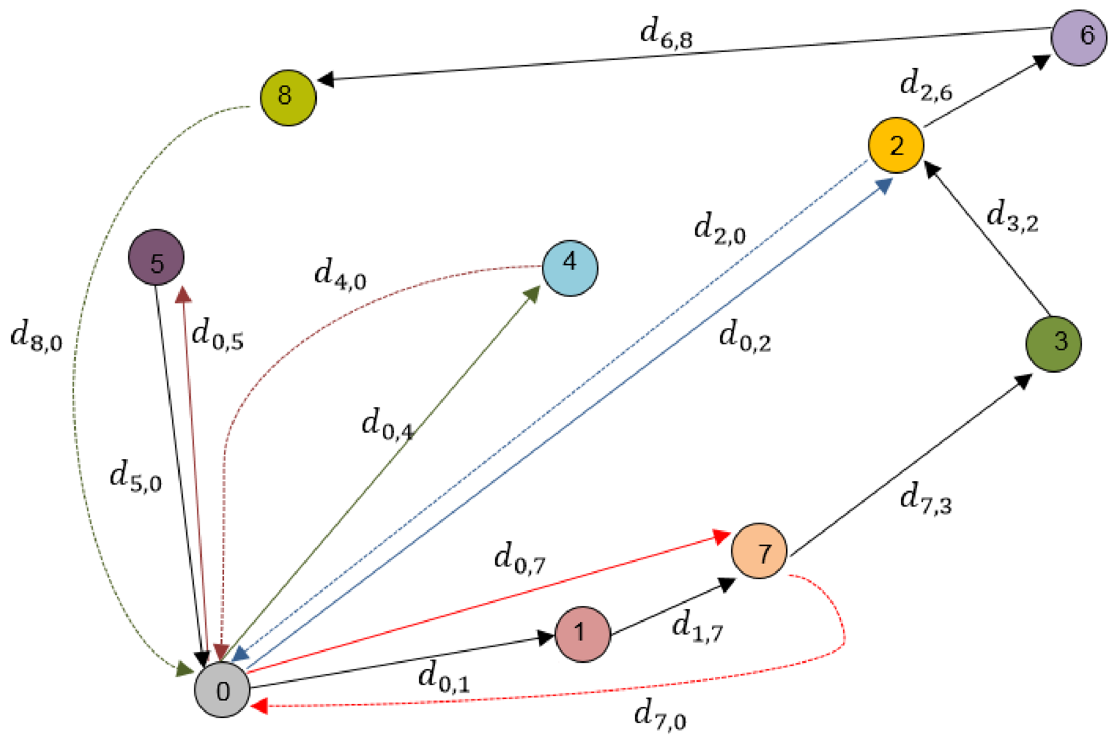
Para el presente problema de 8 clientes, de acuerdo a los controles resultantes, el vehículo realizó cuatro idas al depósito para reabastecerse, dos por fallas de ruta y dos por reabastecimiento preventivo. En la etapa 1 el vehículo se dirige al cliente 7 y este demanda una cantidad mayor que la capacidad residual, presentándose un fallo de ruta, por lo que en la atapa 2 el vehículo debe ir al depósito y volver al cliente 7 para servirlo totalmente. Otro fallo de ruta se percibe en la etapa 5.

El fracaso de una ruta se debe a la aleatoriedad de las demandas, ya que cada cliente tiene cuatro posibles demandas pero solo cuando el vehículo llega a la ubicación del cliente se observa el número de artículos que este necesita pedir.

En la etapa 8 y 9, el vehículo se dirige hacia el cliente 4 y 5 respectivamente, pero pasando antes por el depósito a realizar un reabastecimiento preventivo.

Lo descrito anteriormente se ilustra a continuación en la figura 16:

Figura 16. Ruta final con retornos para 8 clientes



9.4.3 Análisis de controles para doce clientes

- Ruta final: 0 - 10 - 4 - 5 - 12 - 9 - 2 - 11 - 3 - 7 - 1 - 6 - 8 - 0
- Controles resultantes: $\mu_0(10,0), \mu_1(4,0), \mu_2(4,1), \mu_3(5,0), \mu_4(12,0), \mu_5(12,1),$ $\mu_6(9,0), \mu_7(2,0), \mu_8(2,1), \mu_9(6,0), \mu_{10}(8,0), \mu_{11}(7,1), \mu_{12}(1,0), \mu_{13}(1,1),$ $\mu_{14}(3,0), \mu_{15}(11,0), \mu_{16}(0,0)$'

Tabla 22. Posibles demandas para 12 clientes

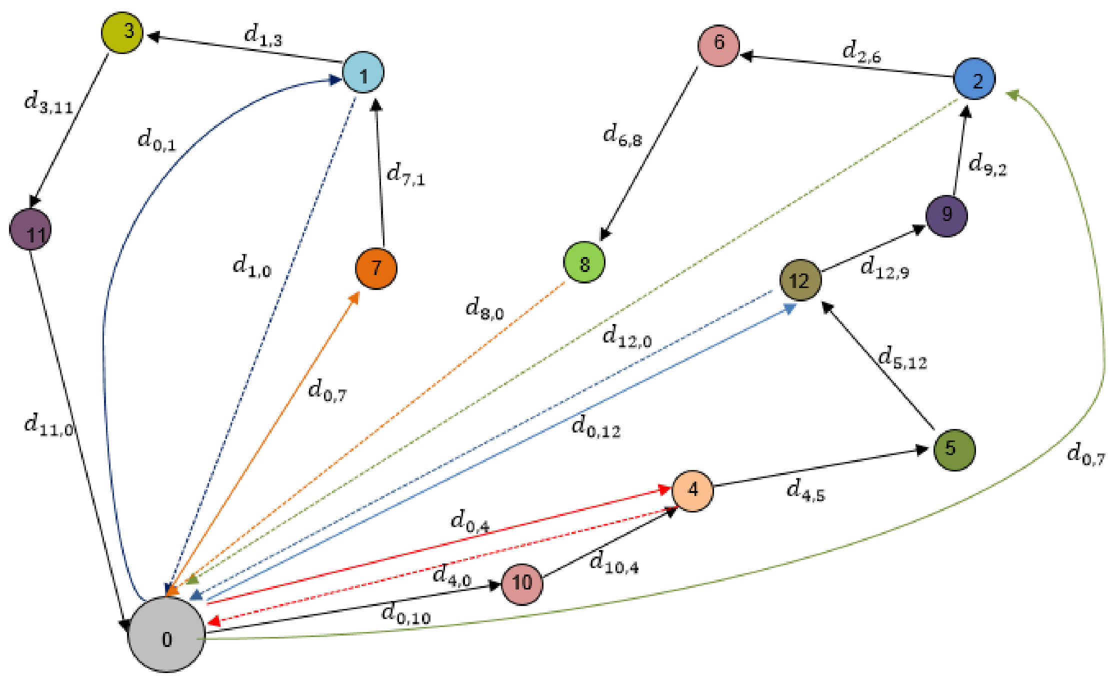
Capacidad (Q) 121	**Demanda 1**		**Demanda 2**		**Demanda 3**		**Demanda 4**	
	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad	Demanda	Probabilidad
Cliente 1	51	0,25	52	0,25	53	0,25	54	0,25
Cliente 2	57	0,25	59	0,25	55	0,25	56	0,25
Cliente 3	58	0,25	56	0,25	57	0,25	63	0,25
Cliente 4	67	0,25	68	0,25	66	0,25	65	0,25
Cliente 5	63	0,25	64	0,25	65	0,25	66	0,25
Cliente 6	57	0,25	53	0,25	51	0,25	54	0,25
Cliente 7	69	0,25	71	0,25	68	0,25	67	0,25
Cliente 8	59	0,25	56	0,25	60	0,25	61	0,25
Cliente 9	55	0,25	52	0,25	51	0,25	53	0,25
Cliente 10	68	0,25	62	0,25	65	0,25	69	0,25
Cliente 11	54	0,25	56	0,25	58	0,25	55	0,25
Cliente 12	71	0,25	69	0,25	63	0,25	64	0,25

Por último, se tiene el problema con 12 clientes. Teniendo en cuenta los controles resultantes, se observa cuatro fallas de ruta en las etapas 2, 5, 8 y 13, el vehículo estando en los clientes 4, 12, 2 y 1 respectivamente, tiene que ir al depósito y volver para suplir la demanda de estos clientes completamente. En la etapa 11 se genera un reabastecimiento preventivo, cuando el vehículo se encuentra en el cliente 8 y se dirige al 7 pero antes pasa por el depósito.

Como se explicaba anteriormente, los fallos de rutas se presentan debido a que las demandas de los clientes son estocásticas o aleatorias y cada cliente puede demandar cuatro posibles valores, lo que hace el problema más complejo. También, se generan más idas al depósito porque el número de clientes es mayor en este caso, teniendo la misma capacidad del problema con 4 clientes.

A continuación en la figura 17 se ilustra el recorrido del vehículo con doce clientes:

Figura 17. Ruta final con retornos para 8 clientes



9.5 VALIDACIÓN DE LA SOLUCIÓN OBTENIDA POR EL SOFTWARE PARA MEDIR LA CALIDAD DE LOS RESULTADOS

Los resultados de la función objetivo obtenidos con el algoritmo Rollout son comparados con el método EACO empleando los mismos factores para ambos problemas, esto se realiza con el fin de validar las soluciones obtenidas y medir la calidad de los resultados empleando el algoritmo Rollout.

En la tabla 23 se muestran los resultados hallados con los dos métodos empleados. Los resultados muestran que de las 24 instancias, el 37,5% indica que el Rollout obtiene mejores resultados que el método EACO. Además se observa que la mayor diferencia porcentual que hay por encima del algoritmo es de 10,5% y el mayor valor por debajo es de 13,7%.

Tabla 23. Resultados del valor de la función objetivo EACO-Rollout

INSTANCIA	EACO	ROLLOUT	DIFERENCIA PROCENTUAL
1	87,1805	89,175226	2,3%
2	103,03808	114,59144	11,2%
3	86,5461	90,451154	4,5%
4	105,04526	118,233002	12,6%
5	89,74948	94,470784	5,3%
6	83,091	90,7667	9,2%
7	104,03002	106,116574	2,0%
8	100,30024	114,02512	13,7%
9	168,70056	167,49716	0,7%
10	177,23872	183,36254	3,5%
11	179,51922	186,74602	4,0%
12	190,041	191,54096	0,8%
13	182,01876	162,87122	10,5%
14	176,76778	190,45842	7,7%
15	183,37802	177,60888	3,1%
16	189,8625	196,64172	3,6%
17	239,98156	234,958	2,1%
18	245,43328	241,38144	1,7%
19	256,70084	246,904	3,8%
20	264,93538	243,47142	8,1%
21	259,491	249,63766	3,8%
22	242,82822	250,07362	3,0%
23	258,6117	243,59108	5,8%
24	258,73526	267,00408	3,2%

10. CONCLUSIONES

Los objetivos planteados para esta investigación se cumplieron en su totalidad. El algoritmo Rollout se desarrolló en Microsoft Excel 2010 para resolver el problema de ruteo de vehículos con demandas estocásticas bajo una perspectiva de reoptimización, arrojando resultados factibles.

A partir de la revisión bibliográfica, se decide tratar el algoritmo Rollout de un solo paso (*one-step rollout algorithm ORA*), el cual es un tipo de regla de iteración de un paso de mejoramiento sobre los estados del sistema, con el objetivo de encontrar una política mejorada π[85].

El Rollout es un algoritmo que parte de una ruta a priori y empleando una heurística cíclica tiene como finalidad optimizar la toma de decisiones (controles) en cada etapa, para finalmente obtener una aproximación al costo óptimo de la ruta resultante.

A partir del resultado del diseño de experimentos se concluye que:

- El análisis del algoritmo se dividió para 4, 8 y 12 clientes. En los efectos estimados, para 4 y 8 clientes la ubicación de los clientes y la demanda promedio respectivamente, son factores significativos sobre la variable de respuesta.

- Para obtener mejores resultados sobre la función objetivo la ubicación de los clientes y la demanda promedio deben tomar el nivel 1 mientras que la desviación debe tomar el nivel 2.
- En el grupo de 12 clientes ningún factor es significativo sobre la variable respuesta.

[85] NOVOA. y STORER, Op. cit., p 511.

- El análisis de varianza muestra que las interacciones entre los factores no tiene influencia en ningún grupo de clientes sobre el costo de la ruta final.

- Se trabaja con parámetros propios del problema de ruteo de vehículo con demandas estocásticas, siendo tres de estos los factores analizados para el algoritmo Rollout.

En cuanto a los controles o decisiones tomadas por el vehículo a lo largo de la trayectoria, se observa que el programa arroja resultados coherentes, realizando los retornos necesarios al depósito. Teniendo en cuenta la aleatoriedad de las demandas de cada cliente, en cada corrida se obtiene controles diferentes y como consecuencia un costo distinto.

Los resultados obtenidos al comparar el Rollout con el método EACO muestran que el 37,5% de las instancias, para el algoritmo Rollout, arroja mejores resultados en el grupo de 8 y 12 clientes, con una diferencia porcentual mayor al 1%.

11. RECOMENDACIONES

- El código de programación del algoritmo se deja a disposición del lector para su estudio, modificación o mejora de tal forma que sirva de base a otras investigaciones.

- Revisar y analizar para futuras investigaciones la posibilidad de desarrollar una herramienta más robusta, que permita trabajar con un número mayor de clientes y reducir el tiempo de cómputo para una aplicación más cercana a casos reales de la industria.

- Incentivar la realización de proyectos de grado que aborden investigaciones relacionadas con la aplicación del algoritmo Rollout, con el objetivo de que este sea utilizado como herramienta para aumentar la eficiencia y competitividad de las organizaciones.

- Desarrollar la herramienta en otro lenguaje de programación más robusto que permita aumentar el número de clientes en las instancias que sirvan como marco de referencia para validar la calidad de los resultados.

- El presente trabajo servirá como base para la realización de futuros proyectos, ya que no existen antecedentes de proyectos de grados en la Universidad Industrial de Santander que utilicen el algoritmo Rollout como solución para el VRPSD.

BIBLIOGRAFÍA

Ak, A y A. Erera. A paired-vehicle routing recourse strategy for the vehicle routing problem with stochastic demands. En: Transportation Science. Vol. 41, No. 2, (2007); p 222–237.

ALFONSECA, Manuel. La máquina de Turing. [en línea]. p 165-168 [Consultado 7 Jun. 2013]. Disponible en: http://www.sinewton.org/numeros/numeros/43-44/Articulo33.pdf

AZI, Nabila; GENDREAU, Michel y POTVIN Jean-Y ves. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. En: European Journal of Operational Research. Vol. 178, (2007); p. 756-763.

BERTSEKAS, Dmitri P. Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC. En: European Journal of Operational Research, (2005); p. 5.

BERTSIMAS, Dimitris; CHERVI, Philippe Y PETERSON, Michael. Computational Approaches to Stochastic Vehicle Routing Problems. En: Transportation Science. Vol. 29, No. 4 (1995); p. 342–352.

BIANCHI, Leonora, et al. Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Journal of Mathematical Modelling and Algorithms. (2006); p. 91-110.

________ Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Parallel Problem Solving from Nature-PPSN. Enero, 2004. vol. 8, p. 450.

CAMPBELL, Ann Melissa. Aggregation for the probabilistic traveling salesman problem. En: Computers and Operations Research,(2006);p 1-22.

CERDÁ, Emilio. Optimización Dinámica. 1 ed. México: Alfaomega Grupo Editor, 2012. p. 215-217.

CHANG, Mei-Shiang. A vehicle routing problem with time windows and stochastic demands. En: Journal of the Chinese Institute of Engineers. Vol. 28, No. 5, (2005.); p.783-794.

CHEPURI, Krishana y HOMEM-DE-MELLO Tito. Solving the Vehicle Routing Problem with Stochastic Demands using the Cross Entropy Method. [en línea].(2003). [Consultado 3 Jun. 2013]. Disponible en: <http://www.optimization-online.org/DB_FILE/2004/06/893.pdf>.

CORDEAU, Jean François., et al. Vehicle Routing. En: Handbooks in Operations Research & Management Science: Transportation. 2007. p.367.

CORTÉZ, Augusto. Teoría de la complejidad computacional y teoría de la computabilidad. En: Rev. investig. sist. inform. Facultad de Ingeniería de Sistemas e Informática. Universidad Nacional Mayor de San Marcos. [en línea]. [consultado 8 junio 2013]. (2004); p. 102-105. Disponible en: <http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/risi/n1_2004/a14.pdf>

C.Y. Cheong, et al. A multiobjetive evolutionary algorithm for solving vehicle routing problem with stochastic demand. En: Congreso sobre computación evolutiva (16-21, Julio, 2006: Vancouver, Canadá). Memorias. Instituto de ingenieros eléctricos y electrónicos- IEEE, 2006. p.1370-1377.

DROR, Mosher; LAPORTE, Gilbet y TRUDEAU, Pierre. Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks. En: Transportation Science. Agosto, 1989. Vol. 23, no.3,. p. 168.

EKSIOGLU, Burak; VOLKAN, Arif Vural y REISMAN, Arnold. The vehicle routing problem: A taxonomic review. En: Computers & Industrial Engineering. Vol. 57, (2009); p, 1473.

ERERA, Alan L; MORALES, Juan C y SAVELSBERGH, Martin. The Vehicle Routing Problem with Stochastic Demand and Duration Constraints. En: The Logistics Institute Georgia Institute of Technology School of Industrial and Systems Engineering Atlanta [en línea]. (mar. 2009). [Consultado 3 Jun. 2013]. Disponible en http://www2.isye.gatech.edu/~mwps/publications/VRPSD_DC_02Mar09.pdf

GALVÁN, Silvia; ARIAS, Javier y LAMOS, Henry. En: DYNA. Junio, 2013. Ed,179. pp. 60-69

GENDREAU, M., LAPORTE, G. y SÉGUIN, R., An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers. En: Transportation Science.1995.pp. 143-155.

GÓMEZ, David y RANGEL, Carlos. Formular las Metaheurísticas Búsqueda Tabú y Recocido Simulado para la solución del CVRP (Capacitated Vehicle Routing Problem). [en línea]. Bucaramanga (2011); 150 h. Trabajo de grado (Ingeniería Industrial). Universidad Industrial de Santander. Facultad de Ingeniería Físico-Mecánicas. [Consultado 7 Jun. 2013]. Disponible en: <http://tangara.uis.edu.co/biblioweb/tesis/2011/137753.pdf>.

GOMEZ ATUESTA, David Fernando y RANGEL CARVAJAL, Carlos Eduardo. Formular las metahurísticas búsqueda tabú y recocido simulado para la solución del CVRP (capacitated vehicle routing problema).Bucaramanga, 2011, 150p. Tesis (Ingeniero Industrial). Universidad Industrial De Santander. Facultad de Ingeniería Fisico-Mecanicas. Escuela de Estudios industriales y Empresariales.

GOODSON, Justin C; THOMAS Barrett W y OHLMANN Jeffrey W. Restocking-Based Rollout Policies for the Vehicle Routing Problem with Stochastic Demand and Duration. En: [en línea]. (mar. 2009). [Consultado 3 Jun. 2013]. Disponible en <http://www.slu.edu/~goodson/papers/GoodsonRestockingVRPSDL.pdf>.

HAMALAINEN, Wilhemeiina. Class NP, NP-complete, and NP-hard problems. [en línea]. (2006). [Consultado 7 Jun. 2013]. Disponible en: https://cs.joensuu.fi/pages/whamalai/daa/npsession.pdf

HILLIER, Frederick y LIEBERMAN, Gerald. Introducción a la Investigación de Operaciones. 9 ed. México: Mc Graw Hill. p. 445-447.

JUAN. A. Angel, et al. Applying Simulation and Reliability to Vehicle Routing Problems with Stochastic Demands. [en línea]. [consultado 25 May. 2013]. Disponible en <http://ceur-ws.org/Vol-589/paper02.pdf>.

LAPORTE, Gilbert; GENDREAU, Michel y HERTZ, Alain. An aproximation algorithm for the traveling salesman problem with time windows. En: Institute for Operation Research and de Management Science. Vol. 45, No. 4, (1998); p 639-641

LAWLER, Eugene. Combinatorial Optimization: Networks and Matroids. [en línea]. Courier Dover, (1976); p 1-4. [Consultado 7 Jun. 2013]. Disponible en: <http://www.plouffe.fr/simon/math/CombinatorialOptimization.pdf>.

LÜER, Armin., et.al. El problema de rutas de vehículos: Extensiones y métodos de resolución, estado del arte. [en línea]. (2009). [Consultado 3 Jun. 2013]. Disponible en: <http://ceur-ws.org/Vol-558/Art_23.pdf>.

MARIUS, Muja y LOWE David G. Fast approximate nearest neighbors with automatic algorithm configuration. En: Computer Science Department, University of British Columbia, Vancouver, B.C., Canada; p 1.

NOVOA, Clara. The Vehicle Routing Problem with Stochastic Demands Combining Static and Dynamic Approaches. p 1-7.

NOVOA, Clara., et al. A Set-Partitioning-Based Model for the Stochastic Vehicle Routing Problem. En: Computational Optimization Research At Lehigh:Technical Report. [en línea]. [consultado 12 junio 2013]. (2006); Disponible en: <http://www.optimization-online.org/DB_FILE/2006/12/1542.pdf>

NOVOA, Clara y STORER, Robert. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. En: European journal of operational research. Vol. 196, No 1 (2009); p 509-515.

OLIVERA Alfredo. Heurísticas para problemas de ruteo de vehículos. En: Instituto de computación, facultad de ingeniería, universidad de la república. [en línea]. Montevideo, Uruguay, Agosto 2004. [consultado 25 may. 2013]. Disponible en <http://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0408.pdf>.

PSARAFTIS, Harilaos. Dynamic vehicle routing: Status and prospects. En: Operations Research. Vol. 61, No 1 (1995); p143-164.

RAMOS, Andres. Programación dinámica. [en línea]; p 25-26. [Consultado 23 Ene. 2014]. Disponible en: < http://www.iit.upcomillas.es/aramos/simio/transpa/t_dp_ar.pdf>.

RIBAS, Sabir., et.al. A hybrid algorithm for the Vehicle Routing Problem with Time Windows. [en línea]. (2011). [Consultado 3 Jun. 2013]. Disponible en: http://www2.ic.uff.br/~satoru/conteudo/artigos/PAPER%20IESM2011-SABIR.pdf

ROCHA, Linda; GONZALEZ, Cristina y ORJUELA, Javier. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol. 16, No 2, (2011); p. 35-55.

SECOMANDI, Nicola. A rollout policy for the vehicle routing problem with stochastic demands. En: Institute for Operation Research and de Management Science. Vol. 49, No. 5, (2001); p 798.

________ Analysis of a Rollout Approach to Sequencing Problems with Stochastic Routing Applications. En: Journal of heuristics. (2003); p. 321-352.

________ Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. En: Computers & Operations Research. Vol. 27. (2000); p. 1201-1225.

SHEN, Zhihong; ORDÓÑEZ, Fernando y DESSOUKY, Maged M. The Stochastic Vehicle Routing Problem for Minimum Unmet Demand. En: Optimization and Logistics Challenges in the Enterprise Springer Optimization and Its Applications. 2009. vol, 30.p 350.

SHEN, Lin. Computer Solutions of the Traveling Salesman Problem. [en línea]. (Agosto 18, 1965). [Consultado 3 Jun. 2013]. Disponible en: <http://www3.alcatel-lucent.com/bstj/vol44-1965/articles/bstj44-10-2245.pdf>

TAN, K.C; CHEONG, C.Y y GOH C.K.. En: European Journal of Operational Research. Vol. 177, (2007); p. 813-839.

TATARAKIS, A y MINIS, I. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. En: European Journal of Operational Research. Vol. 34, (2009); p. 558.

YANG, Wen-Huei; MATHUR, Kamlesh y BALLOU, Ronald. Stochastic vehicle routing problem with restocking. En: Transportation science. Vol. 34, No. 1, (2000); p 99-102.

YOUNG, Peng y HA-YING, Zhu. Research on Vehicle Routing Problem with Stochastic Demandand PSO-DP Algorithm with Inver-over Operator. En: Systems Engineering- Theory & Practice. Vol. 10, (2008); p. 76-81.

ANEXOS

ANEXO A. CÓDIGO DEPROGRAMACIÓN

```
Dim nivel As Integer

Dim clientes As Integer
Dim demandas As Integer

Dim controlesOptimizados() As Integer
Dim indControles As Integer

Dim matrizDistancias() As Integer
Dim matrizDemandas() As Double

Dim rutaPriori() As Integer
Dim rutaPrioriText As String
Dim rutaFinal() As Integer
Dim rutaTemporal() As Integer

Dim costoRuta As Double

Dim Q As Integer

Sub ejecutar()

    Dim errorDatos As Boolean

    Dim ti As Double
    Dim tf As Double

    Dim acumularProbabilidad As Double

    errorDatos = False

    ThisWorkbook.Sheets(1).Cells.Clear
    If ThisWorkbook.Sheets(1).ChartObjects.Count > 0 Then
        ThisWorkbook.Sheets(1).ChartObjects.Delete
    End If
    ThisWorkbook.Sheets(1).Range("G2").Value = "Ruta a Priori:"
    ThisWorkbook.Sheets(1).Range("G3").Value = "Ruta Final:"
    ThisWorkbook.Sheets(1).Range("G5").Value = "Controles a optimizar: "
    ThisWorkbook.Sheets(1).Range("H4").Value = "Ruta Priori"
    ThisWorkbook.Sheets(1).Range("I4").Value = "Ruta Final"
    ThisWorkbook.Sheets(1).Range("A12").Value = "Costo Ruta a Priori:"
    ThisWorkbook.Sheets(1).Range("A13").Value = "Costo Ruta Final:"
```

```vb
    ThisWorkbook.Sheets(1).Range("K3").Value = "Matriz Distancias:"
    ThisWorkbook.Sheets(1).Range("A15").Value = "Tiempo de Ejecución (s):"

    'comprobamos que hayan ingresado el número de clientes y demandas

    '***********************************************************
    'MsgBox "Comprobaciones de rigor para verificación de datos"
    '***********************************************************

    If ThisWorkbook.Sheets(2).Range("B1").Value = "" Then
        MsgBox "Debe ingresar el numero de clientes en la hoja de Datos, celda
B1"
        errorDatos = True
    Else
        clientes = ThisWorkbook.Sheets(2).Range("B1").Value
        ReDim matrizDistancias(clientes, 2)
        ReDim rutaPriori(clientes + 2)
        ReDim rutaFinal(clientes + 2)
        ReDim rutaTemporal(clientes + 2)
        ReDim controlesOptimizados(40, 2)

        For ind = 0 To clientes
            ThisWorkbook.Sheets(1).Cells(4, 12 + ind).Value = ind
            ThisWorkbook.Sheets(1).Cells(5 + ind, 11).Value = ind
        Next ind

        'ahora comprobamos que hayan ingresado las demandas
        If ThisWorkbook.Sheets(2).Range("B2").Value = "" Then
            MsgBox "Debe ingresar el numero de demandas, en la hoja de Datos,
celda B2"
            errorDatos = True
        Else
            demandas = ThisWorkbook.Sheets(2).Range("B2").Value
            ReDim matrizDemandas(clientes, demandas, 2)
            'comprobamos que hayan escrito la Q
            If ThisWorkbook.Sheets(2).Range("D1").Value = "" Then
                MsgBox "Debe ingresar la capacidad del vehiculo, en la hoja de
Datos, celda D1"
                errorDatos = True
            Else
                Q = ThisWorkbook.Sheets(2).Range("D1").Value
                'se comprueba que este correctamente llenada la matriz de
                 distancias de clientes y ademas que no se repitan
                For ind = 0 To clientes - 1
                    If ThisWorkbook.Sheets(2).Range("B" & 6 + ind).Value <> ""
Or ThisWorkbook.Sheets(2).Range("C" & 6 + ind).Value <> "" Then
```

```
                        matrizDistancias(ind, 1) =
ThisWorkbook.Sheets(2).Range("B" & 6 + ind).Value
                        matrizDistancias(ind, 2) =
ThisWorkbook.Sheets(2).Range("C" & 6 + ind).Value
                    Else
                        MsgBox "Ha dejado un cliente sin diligenciar, por favor
revise la hoja Datos para corregir el error"
                        errorDatos = True
                        Exit For
                    End If
                Next ind
                'se han obtenido todos los valores, verificamos que no estén
                 repetidos
                For ind = 0 To clientes - 2
                    For jnd = ind + 1 To clientes - 1
                        If matrizDistancias(ind, 1) = matrizDistancias(jnd, 1)
And matrizDistancias(ind, 2) = matrizDistancias(jnd, 2) Then
                            MsgBox "Los clientes no pueden tener las mismas
coordenadas, por favor corregir en la hoja de Datos"
                            errorDatos = True
                            Exit For
                        End If
                    Next jnd
                Next ind
                'ahora se verifican las demandas, que esten completas en
                 clientes y en total de demandas
                For ind = 0 To clientes - 1
                    acumularProbabilidad = 0
                    For jnd = 0 To demandas - 1
                        If ThisWorkbook.Sheets(2).Cells(7 + ind, 6 + 2 *
jnd).Value <> "" And _
                           ThisWorkbook.Sheets(2).Cells(7 + ind, 7 + 2 *
jnd).Value <> "" Then
                            matrizDemandas(ind, jnd, 1) =
ThisWorkbook.Sheets(2).Cells(7 + ind, 6 + 2 * jnd).Value
                            matrizDemandas(ind, jnd, 2) =
ThisWorkbook.Sheets(2).Cells(7 + ind, 7 + 2 * jnd).Value
                            acumularProbabilidad = acumularProbabilidad +
matrizDemandas(ind, jnd, 2)
                        Else
                            MsgBox "No ha diligenciado correctamente las
demandas de los clientes. Por favor verifique esta informacion en la" _
                                    & " hoja Datos"
                            errorDatos = True
                            Exit For
                        End If
```

```
                    Next jnd
                    If errorDatos = True Then
                        Exit For
                    Else
                        If acumularProbabilidad <> 1 Then
                            MsgBox "La suma de probabilidades de demanda del
cliente " & ind + 1 & " no es 1, por favor rectifique" & _
                                " los valores en la hoja Datos"
                            errorDatos = True
                            Exit For
                        End If
                    End If
                Next ind
            End If
        End If
    End If

    If errorDatos = False Then

        '***********************************************************
        'MsgBox "Graficamos los puntos"
        '***********************************************************

        'creamos la gráfica que representa los puntos coordenados de los
         clientes
        graficarPuntos

        'insertar distancias
        insertarDistancias

        '***********************************************************
        'MsgBox "Calculamos la ruta a priori"
        '***********************************************************

        'se ha terminado de verificar si hay errores, se procede con el
        algoritmo, se calcula la ruta a priori
        calcularRutaPriori
        ThisWorkbook.Sheets(1).Range("H2").Value = rutaPrioriText

        '***********************************************************
        'MsgBox "Se ubican los controles Uk a priori que seran actualizados"
        '***********************************************************

        'se ubican los controles u_k de la ruta a priori?
        controlesUkPriori
```

```
        '*********************************************************
        'MsgBox "Se calcula el costo de la ruta a priori actual (ni idea para
         que)"
        '*********************************************************

        'calculamos el costo de la ruta
        calcularCostoRuta
        ThisWorkbook.Sheets(1).Range("B12").Value = costoRuta

        '*********************************************************
        'MsgBox "Se calcula la ruta, Q: " & Q
        '*********************************************************

        'ahora se procede a calcular el cambio de paso
        indControles = 0
        ti = Timer
        calcularRuta Q
        tf = Timer
        ThisWorkbook.Sheets(1).Range("B15").Value = tf - ti

        'se imprimie la ruta final
        rutaPrioriText = "0"
        For ind = 1 To clientes
            rutaPrioriText = rutaPrioriText & " - " & rutaFinal(ind)
        Next ind
        rutaPrioriText = rutaPrioriText & " - 0"
        ThisWorkbook.Sheets(1).Range("H3").Value = rutaPrioriText

        'ahora se procede a calcular el costo de la ruta final
        calcularCostoRutaFinal

    End If

End Sub

Public Sub graficarPuntos()

    Dim grafica As ChartObject
    Dim area As Range
    Dim datos As Range

    Set area = ThisWorkbook.Sheets(1).Range("A1:F10")
    Set datos = ThisWorkbook.Sheets(2).Range("B6:C" & 5 + clientes)

    Set grafica = ThisWorkbook.Sheets(1).ChartObjects.Add( _
                    Left:=area.Left, Top:=area.Top, _
```

```
                    Width:=area.Width, Height:=area.Height)

    With grafica.Chart
        .ChartType = xlXYScatter
        .SetSourceData Source:=datos
        .HasTitle = False
    End With

End Sub

Public Sub insertarDistancias()

    Dim distancia As Double

    For ind = 0 To clientes
        ThisWorkbook.Sheets(1).Cells(5 + ind, 12 + ind).Value = 0
        For jnd = ind + 1 To clientes
            If ind = 0 Then
                distancia = Sqr( _
                               matrizDistancias(jnd - 1, 1) ^ 2 _
                               + matrizDistancias(jnd - 1, 2) ^ 2 _
                                )
            Else
                distancia = Sqr( _
                                (matrizDistancias(jnd - 1, 1) -
matrizDistancias(ind - 1, 1)) ^ 2 _
                                + (matrizDistancias(jnd - 1, 2) -
matrizDistancias(ind - 1, 2)) ^ 2 _
                                 )
            End If
            ThisWorkbook.Sheets(1).Cells(5 + ind, 12 + jnd).Value = distancia
            ThisWorkbook.Sheets(1).Cells(5 + jnd, 12 + ind).Value = distancia
        Next jnd

    Next ind


End Sub

Public Sub calcularRutaPriori()

    Dim distancia As Double
    Dim distanciaCalculada As Double
    Dim vectorClientes() As Integer

    ReDim vectorClientes(clientes)
```

```
    For ind = 0 To clientes - 1
        vectorClientes(ind) = ind + 1
    Next ind

    'asignamos la bodega
    rutaPriori(0) = 0
    rutaPrioriText = "0"

    'buscamos la ruta más cercana
    distancia = 1E+21
    For ind = 1 To clientes
        For jnd = 0 To clientes - 1
            If ind = 1 Then
                distanciaCalculada = Sqr(matrizDistancias(jnd, 1) ^ 2 +
matrizDistancias(jnd, 2) ^ 2)
            Else
                'For knd = 1 To ind - 1
                    If vectorClientes(jnd) <> 0 Then
                        distanciaCalculada = Sqr((matrizDistancias(jnd, 1) -
matrizDistancias(rutaPriori(ind - 1) - 1, 1)) ^ 2 _
                                                + (matrizDistancias(jnd, 2) -
matrizDistancias(rutaPriori(ind - 1) - 1, 2)) ^ 2)
                    End If
                'Next knd
            End If
            If distanciaCalculada < distancia And distanciaCalculada <> 0 Then
                distancia = distanciaCalculada
                rutaPriori(ind) = jnd + 1
            End If
        Next jnd
        vectorClientes(rutaPriori(ind) - 1) = 0
        rutaPrioriText = rutaPrioriText & " - " & rutaPriori(ind)
        distancia = 1E+21
        distanciaCalculada = 1E+21 + 1
    Next ind

    rutaPriori(clientes + 2) = 0
    rutaPrioriText = rutaPrioriText & " - 0"

End Sub

Public Sub controlesUkPriori()

    For ind = 1 To clientes
        ThisWorkbook.Sheets(1).Cells(4 + ind, 8).Value = "u" & (ind - 1) & "(" &
rutaPriori(ind) & ", 0)"
```

```
    Next ind
    ThisWorkbook.Sheets(1).Cells(4 + clientes + 1, 8).Value = "u" & (clientes) &
"(0, 0)"

End Sub

Public Sub calcularCostoRuta()

    Dim costo As Double

    For ind = 1 To clientes + 1
        If ind = 1 Then
            costoRuta = Sqr((matrizDistancias(rutaPriori(ind) - 1, 1)) ^ 2 +
(matrizDistancias(rutaPriori(ind) - 1, 2)) ^ 2)
        Else
            If ind = clientes + 1 Then
            costo = Sqr((matrizDistancias(rutaPriori(ind - 1) - 1, 1)) ^ 2 _
                        + (matrizDistancias(rutaPriori(ind - 1) - 1, 2)) ^ 2)
            costoRuta = costoRuta + costo
            Else
                costo = Sqr((matrizDistancias(rutaPriori(ind) - 1, 1) -
matrizDistancias(rutaPriori(ind - 1) - 1, 1)) ^ 2 _
                        + (matrizDistancias(rutaPriori(ind) - 1, 2) -
matrizDistancias(rutaPriori(ind - 1) - 1, 2)) ^ 2)
                costoRuta = costoRuta + costo
            End If
        End If
    Next ind

End Sub

Public Sub calcularCostoRutaFinal()

    Dim costoTotal As Double

    For ind = 0 To indControles - 1
        ThisWorkbook.Sheets(1).Cells(5 + ind, 9).Value = "u" & (ind) & "(" &
controlesOptimizados(ind, 1) & ", " & _

controlesOptimizados(ind, 2) & ")"

        If ind = 0 Then
            costoTotal = costoTotal +
Sqr((matrizDistancias(controlesOptimizados(ind, 1) - 1, 1)) ^ 2 _
                            + (matrizDistancias(controlesOptimizados(ind, 1) -
1, 2)) ^ 2)
```

```
        Else
            If controlesOptimizados(ind, 2) = 0 Then
                'va directamente al cliente
                costoTotal = costoTotal +
Sqr((matrizDistancias(controlesOptimizados(ind, 1) - 1, 1) _
                                                -
matrizDistancias(controlesOptimizados(ind - 1, 1) - 1, 1)) ^ 2 _
                                            +
(matrizDistancias(controlesOptimizados(ind, 1) - 1, 2) _
                                                -
matrizDistancias(controlesOptimizados(ind - 1, 1) - 1, 2)) ^ 2)
            Else
                'va al depósito, luego al cliente
                costoTotal = costoTotal +
Sqr((matrizDistancias(controlesOptimizados(ind - 1, 1) - 1, 1)) ^ 2 _
                            + (matrizDistancias(controlesOptimizados(ind - 1, 1)
- 1, 2)) ^ 2) _
                            + Sqr((matrizDistancias(controlesOptimizados(ind, 1)
- 1, 1)) ^ 2 _
                            + (matrizDistancias(controlesOptimizados(ind, 1) -
1, 2)) ^ 2)
            End If
        End If
    Next ind

    ThisWorkbook.Sheets(1).Cells(5 + ind, 9).Value = "u" & indControles & "(0,
0)"
    costoTotal = costoTotal +
Sqr((matrizDistancias(controlesOptimizados(indControles - 1, 1) - 1, 1)) ^ 2 _
                            +
(matrizDistancias(controlesOptimizados(indControles - 1, 1) - 1, 2)) ^ 2)

    ThisWorkbook.Sheets(1).Range("B13").Value = costoTotal

End Sub

Public Sub calcularRuta(ByVal Qi As Integer)

    Dim tempV As Integer
    Dim textRuta As String
    Dim Qb() As Double
    Dim QbMin As Double
    Dim QbMinIndex As Integer
    Dim dRandom As Integer

    Dim qTemp As Double
```

```
Dim control As Integer
Dim controlMinRuta As Integer
Dim cuentaRutas As Integer
Dim tempRutas() As Integer

Dim costoTotal As Double

ReDim Qb(clientes + 2)
ReDim tempRutas(clientes, clientes + 2) As Integer


'***********************************************************
'msgbox "La ruta final se hace igual a la ruta a priori"
'***********************************************************

'hago una copia de la ruta a priori en la ruta final
'For ind = 0 To clientes + 2
    rutaFinal = rutaPriori
'Next ind

'primero, debo de recorrer todos los clientes en intercambios de a 1

'***********************************************************
'msgbox "Vamos a entrar en el for, para hacer las rutas respectivas"
'***********************************************************

For ind = 0 To clientes - 2

    '***********************************************************
    'msgbox "calcularRuta, ind: " & ind
    '***********************************************************

    '***********************************************************
    'msgbox "calcularRuta, ind: " & ind & ", rutaTemporal=rutaFinal"
    '***********************************************************

    rutaTemporal = rutaFinal

    '***********************************************************
    'msgbox "calcularRuta, ind: " & ind & ", calculo demanda aleatoria"
    '***********************************************************

    'ahora, debo generar una demanda aleatoria para escoger entre, y de allí
     calcular mi q residual
    dRandom = Int((demandas) * Rnd)
    'MsgBox "Demanda aleatoria: " & (dRandom + 1)
```

```
If indControles > 0 Then
    If controlesOptimizados(indControles - 1, 2) = 1 Then
        'paso por el deposito, Qi se actualiza para obtener el total de
         la capacidad
        Qi = Q
    End If
End If
Qi = Qi - matrizDemandas(rutaTemporal(ind + 1) - 1, dRandom, 1)
If Qi < 0 Then
    Qi = Qi + Q
    controlesOptimizados(indControles, 1) = rutaTemporal(ind + 1)
    controlesOptimizados(indControles, 2) = 1
    indControles = indControles + 1
End If

'**********************************************************
'MsgBox "calcularRuta, ind: " & ind & ", dRandom: " & dRandom & ", Qi="
 & Qi & ", matrizDemandas(rutaTemporal(ind + 1) - 1, dRandom, 1): " _
        & matrizDemandas(rutaTemporal(ind + 1) - 1, dRandom, 1)
'**********************************************************

'**********************************************************
'msgbox "calcularRuta, ind: " & ind & ", Entrare en el for para calcular
 cada ruta"
'**********************************************************

'ahora, debo intercambiar los clientes, a medida que los voy necesitando
 y/o usando
cuentaRutas = 0
QbMinIndex = 0
For jnd = ind + 2 To clientes

    '**********************************************************
    'msgbox "calcularRuta, jnd: " & jnd
    '**********************************************************

    '**********************************************************
    'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", creamos
     el vector de la ruta temporal nueva"
    '**********************************************************
    tempV = rutaTemporal(ind + 2)
    rutaPrioriText = "0 - " & rutaTemporal(ind + 1)
    For knd = ind + 2 To clientes - 1
        rutaTemporal(knd) = rutaTemporal(knd + 1)
        rutaPrioriText = rutaPrioriText & " - " & rutaTemporal(knd)
    Next knd
```

```
rutaTemporal(clientes) = tempV
rutaPrioriText = rutaPrioriText & " - " & tempV & " - 0"

'*********************************************************
'MsgBox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ",
 rutaTemporal: " & rutaPrioriText
'*********************************************************

'TODO: genero el costo inicial de la ruta, si partimos de la bodega
If ind = 0 And jnd = 2 Then

    '*********************************************************
    'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", debo
     inicializar el costo de la ruta actual"
    '*********************************************************

    'he partido de la bodega, genero costo inicial
    costoTotal = Sqr( _
                    (matrizDistancias(rutaTemporal(1) - 1, 1)) ^
2 + _
                    (matrizDistancias(rutaTemporal(1) - 1, 2)) ^
2 _
                )
    controlesOptimizados(indControles, 1) = rutaTemporal(1)
    controlesOptimizados(indControles, 2) = 0
    indControles = indControles + 1
End If

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", Se
 calcula la distancia menor dependiendo de si debo ir o no a bodega"
'*********************************************************

qTemp = calcularDistancias(ind + 2, Qi, control)
Qb(cuentaRutas) = qTemp

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", qTemp: "
 & qTemp & ", cuentaRutas: " & cuentaRutas
'*********************************************************

'guardamos la ruta de menor costo encontrada hasta ahora
If jnd = ind + 2 Then

'*********************************************************
```

```
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", Es la
 primera ruta calculada, la asumimos minima"
'*********************************************************

    QbMin = qTemp
    QbMinIndex = cuentaRutas
    controlMinRuta = control

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", QbMin: "
 & QbMin & ", QbMinIndex: " & QbMinIndex
'*********************************************************

Else

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ",
 Verificamos que el resultado sea menor al ya guardado"
'*********************************************************

    If QbMin > qTemp Then

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", El
 resultado es menor al ya calculado."
'*********************************************************

        QbMin = qTemp
        QbMinIndex = cuentaRutas
        controlMinRuta = control

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", QbMin: "
 & QbMin & ", QbMinIndex: " & QbMinIndex
'*********************************************************

    End If
End If

'*********************************************************
'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ", se
 incrementa el cuentaRutas"
'*********************************************************

cuentaRutas = cuentaRutas + 1
```

```
            '******************************************************
            'msgbox "calcularRuta, ind: " & ind & ", jnd: " & jnd & ",
             cuentaRutas: " & cuentaRutas
            '******************************************************

            'TODO: dio un resultado menor que el esperado? debo guardar entonces
             la Q residual que quedará para el siguiente ciclo

        Next jnd

            '******************************************************
            'msgbox "calcularRuta, ind: " & ind & ", se actualiza la ruta para
             saber cual es la que esta dando el menor costo"
            '******************************************************

        'TODO: actualizar la ruta final para que quede con la ruta cuyo costo
         dió menor
        rutaTemporal = rutaFinal
        'lo hacemos hasta el indice que encontró era el minimo
        For jnd = ind + 2 To ind + 2 + QbMinIndex
            tempV = rutaTemporal(ind + 2)
            rutaPrioriText = "0 - " & rutaTemporal(ind + 1)
            For knd = ind + 2 To clientes - 1
                rutaTemporal(knd) = rutaTemporal(knd + 1)
                rutaPrioriText = rutaPrioriText & " - " & rutaTemporal(knd)
            Next knd
            rutaTemporal(clientes) = tempV
            rutaPrioriText = rutaPrioriText & " - " & tempV & " - 0"
        Next jnd
        'actualizo la ruta final
        rutaFinal = rutaTemporal
        costoTotal = costoTotal + QbMin
        controlesOptimizados(indControles, 1) = rutaFinal(ind + 2)
        controlesOptimizados(indControles, 2) = controlMinRuta
        indControles = indControles + 1

            '******************************************************
            'msgbox "calcularRuta, ind: " & ind & ", se termina el ciclo del ind
             para empezar uno nuevo (si hay), costoTotal: " & costoTotal & _
                    ", rutaFinal: " & rutaPrioriText
            '******************************************************

    Next ind

    'ahora que estoy en el cliente final, decido si me alcanza, o si debo ir a
     la bodega
```

```
    dRandom = Int((demandas) * Rnd)
    'MsgBox "Demanda aleatoria: " & (dRandom + 1)
    If indControles > 0 Then
        If controlesOptimizados(indControles - 1, 2) = 1 Then
            'paso por el deposito, Qi se actualiza para obtener el total de la
             capacidad
            Qi = Q
        End If
    End If
    Qi = Qi - matrizDemandas(rutaTemporal(ind + 1) - 1, dRandom, 1)
    If Qi < 0 Then
        Qi = Qi + Q
        controlesOptimizados(indControles, 1) = rutaTemporal(ind + 1)
        controlesOptimizados(indControles, 2) = 1
        indControles = indControles + 1
    End If
            '**********************************************************
            'msgbox "calcularRuta, Se acabo la funcion, costoTotal: " &
             costoTotal
            '**********************************************************

End Sub

Public Function calcularDistancias(ByVal indCLientes As Integer, ByVal Qi As
Integer, ByRef control As Integer) As Double

    'creamos el vector donde guardaremos las q residuales
    Dim qia() As Double
    Dim qTemp As Double
    Dim qResult As Double

    Dim formula1 As Double
    Dim formula2 As Double

    Dim controli As Integer

    ReDim qia(Q)

    Dim yaCalculado As Integer
    Dim yaCalculadoResult As Double
    Dim yaCalculadoControl As Integer
    Dim esteNivel As Integer
    Dim preFormula2 As Double
    Dim preControlFormula2 As Integer
```

```
    yaCalculado = 0
    esteNivel = 0

    nivel = nivel + 1
    'MsgBox "calcularDistancias, nivel: " & nivel

            '**********************************************************
            'msgbox "calcularDistancias, entro en la funcion"
            '**********************************************************


            '**********************************************************
            'msgbox "calcularDistancias, verifico si ya es el ultimo cliente
             visitado o no"
            '**********************************************************

    'empiezo por evaluar las demandas residuales del cliente siguiente, a no ser
     que estemos al final del camino, entonces calculo directo
    If indCLientes = clientes + 1 Then
            '**********************************************************
            'msgbox "calcularDistancias, es el ultimo cliente visitado, se
             calcula con la formula 3"
            '**********************************************************
        'calculo directamente la distancia del cliente al depósito
        calcularDistancias = Sqr( _
                                    (matrizDistancias(rutaTemporal(indCLientes -
1) - 1, 1)) ^ 2 + _
                                    (matrizDistancias(rutaTemporal(indCLientes -
1) - 1, 2)) ^ 2 _
                                 )
    Else

            '**********************************************************
            'msgbox "calcularDistancias, estamos visitando un cliente, visitamos
             por cada una de las demandas"
            '**********************************************************

        'calculo las demandas residuales que se generaran por cada formula
        For knd = 0 To demandas - 1

            '*************************************************************
            'formula 1
            '*************************************************************
            If (matrizDemandas(rutaTemporal(indCLientes) - 1, knd, 1) <= Qi)
Then
```

```
                qTemp = Qi - matrizDemandas(rutaTemporal(indCLientes) - 1, knd,
1)
            Else
                qTemp = Q + Qi - matrizDemandas(rutaTemporal(indCLientes) - 1,
knd, 1)
            End If

            'solo si no ha calculado el valor al final de la ruta

            'entro a calcular por cada demanda el valor mínimo del costo de la
             misma
            If qia(qTemp) = Empty Then
                'no ha sido asignado valor, debo verificar lo que le asignaré,
                 será menor?
                If yaCalculado = 0 Then
                    qResult = calcularDistancias(indCLientes + 1, qTemp,
controli)

                    If indCLientes = clientes Then
                        yaCalculado = 1
                        yaCalculadoResult = qResult
                        yaCalculadoControl = controli
                    End If
                ElseIf indCLientes = clientes And yaCalculado = 1 Then
                    qResult = yaCalculadoResult
                    controli = yaCalculadoControl
                End If
                qia(qTemp) = qResult
            End If

            '**************************************************************
            'formula 2
            '**************************************************************
            qTemp = Q - matrizDemandas(rutaTemporal(indCLientes) - 1, knd, 1)

            If esteNivel = 0 Then
                'entro a calcular por cada demanda el valor mínimo del costo de
                 la misma
                qResult = calcularDistancias(indCLientes + 1, qTemp, controli)
                If qia(qTemp) = Empty Or qia(qTemp) > qResult Then
                    'no ha sido asignado valor, debo verificar lo que le
                     asignaré, será menor?
                    qia(qTemp) = qResult
                End If
                esteNivel = nivel
                preFormula2 = qResult
                preControlFormula2 = controli
```

```
            ElseIf esteNivel = nivel Then
                If qia(qTemp) = Empty Or qia(qTemp) > preFormula2 Then
                    'el valor es mayor, lo asignaré
                    qia(qTemp) = preFormula2
                End If
            End If

        Next knd

        'tengo los valores para cada posible demanda y la funcion, debo ahora
         realizar la sumatoria en cada función y decidir qué es menor
        formula1 = Sqr( _
                        ( _
                            matrizDistancias(rutaTemporal(indCLientes - 1) - 1,
1) - _
                            matrizDistancias(rutaTemporal(indCLientes) - 1, 1) _
                        ) ^ 2 + _
                        ( _
                            matrizDistancias(rutaTemporal(indCLientes - 1) - 1,
2) - _
                            matrizDistancias(rutaTemporal(indCLientes) - 1, 2) _
                        ) ^ 2 _
                    )
        formula2 = Sqr( _
                        (matrizDistancias(rutaTemporal(indCLientes - 1) - 1, 1))
^ 2 + _
                        (matrizDistancias(rutaTemporal(indCLientes - 1) - 1, 2))
^ 2) + _
                    Sqr( _
                        (matrizDistancias(rutaTemporal(indCLientes) - 1, 1)) ^ 2
+ _
                        (matrizDistancias(rutaTemporal(indCLientes) - 1, 2)) ^
2)

            '**********************************************************
            'msgbox "calcularDistancias, formula1: " & formula1 & ", formula2. "
             & formula2
            '**********************************************************


            '**********************************************************
            'msgbox "calcularDistancias, entramos en un nuevo ciclo knd de
             demandas"
            '**********************************************************

        For knd = 0 To demandas - 1
```

```
            '**********************************************************
            'msgbox "calcularDistancias, knd: " & knd
            '**********************************************************

            '**********************************************************
            'msgbox "calcularDistancias, knd: " & knd & ", aplicare la formula
             1, determino si un lado o el otro de la sumatoria"
            '**********************************************************

            '*************************************************************
            'formula 1
            '*************************************************************
            If (matrizDemandas(rutaTemporal(indCLientes) - 1, knd, 1) <= Qi)
Then
                qTemp = Qi - matrizDemandas(rutaTemporal(indCLientes) - 1, knd,
1)
                formula1 = formula1 + qia(qTemp) *
matrizDemandas(rutaTemporal(indCLientes) - 1, knd, 2)

            '**********************************************************
            'msgbox "calcularDistancias, knd: " & knd & ", aplico el lado de la
             sumatoria en el que la demanda es menor que la residual"
            '**********************************************************

            '**********************************************************
            'msgbox "calcularDistancias, knd: " & knd & ", qTemp: " & qTemp & ",
             formula1: " & formula1
            '**********************************************************

            Else
                qTemp = Q + Qi - matrizDemandas(rutaTemporal(indCLientes) - 1,
knd, 1)
                formula1 = formula1 + _
                            (2 * Sqr( _

(matrizDistancias(rutaTemporal(indCLientes) - 1, 1)) ^ 2 + _

(matrizDistancias(rutaTemporal(indCLientes) - 1, 2)) ^ 2 _
                                    ) + _
                            qia(qTemp)) * _
                            matrizDemandas(rutaTemporal(indCLientes) - 1, knd,
2)

            '**********************************************************
```

```
'msgbox "calcularDistancias, knd: " & knd & ", aplico el lado de la
 sumatoria en el que la demanda es mayor que la residual"
'*********************************************************

'*********************************************************
'msgbox "calcularDistancias, knd: " & knd & ", qTemp: " & qTemp & ",
formula1: " & formula1
'*********************************************************

End If

'************************************************************
'formula 2
'************************************************************

'*********************************************************
'msgbox "calcularDistancias, knd: " & knd & ", aplico la formula 2"
'*********************************************************

qTemp = Q - matrizDemandas(rutaTemporal(indCLientes) - 1, knd, 1)
            formula2 = formula2 + qia(qTemp) *
matrizDemandas(rutaTemporal(indCLientes) - 1, knd, 2)

'*********************************************************
'msgbox "calcularDistancias, knd: " & knd & ", qTemp: " & qTemp & ",
 formula2: " & formula2
'*********************************************************

Next knd

'*********************************************************
'msgbox "calcularDistancias, salimos del ciclo knd de demandas
 segundo"
'*********************************************************

'*********************************************************
'msgbox "calcularDistancias, evaluamos cual de las dos formulas
 genero el menor costo"
'*********************************************************

'ahora evaluo quien fue menor, y así determino el resultado de la
formula
If formula1 <= formula2 Then

'*********************************************************
```

```
            'msgbox "calcularDistancias, la formula 1 es menor, directamente dirigiendose al cliente"
            '**********************************************************

            calcularDistancias = formula1
            control = 0

            '**********************************************************
            'msgbox "calcularDistancias, formula 1:" & formula1
            '**********************************************************

        Else

            '**********************************************************
            'msgbox "calcularDistancias, la formula 2 es menor, dirigiéndose al deposito y luego al cliente"
            '**********************************************************

            calcularDistancias = formula2
            control = 1

            '**********************************************************
            'msgbox "calcularDistancias, formula 2:" & formula2
            '**********************************************************

        End If

    End If

            '**********************************************************
            'msgbox "calcularDistancias, se acabó la funcion, devolvemos"
            '**********************************************************

    nivel = nivel - 1


End Function


            'imprimo para verificar
            'textRuta = ""
            'For knd = 0 To clientes + 1
            '    If knd <> 0 Then
            '        textRuta = textRuta & " - "
            '    End If
            '    textRuta = textRuta & rutaTemporal(knd)
```

```vb
            'Next knd
            'MsgBox textRuta

        'realizo los calculos respectivos de las demandas de cada cliente
        'If indCLientes = clientes Then
        '    'voy para el depósito, aplico la fórmula dada
        'Else
                '    'no es para el depósito, aplico la fórmula normal, es
        suficiente la demanda?
        '    If Qi <= matrizDemandas(rutaTemporal(indCLientes) - 1, knd) Then
        '        'se aplica una parte de la formula
                '        'se calcula la formula, ahora se llama otra vez esta
        función para que calcule el siguiente cilo
        '        calcularDistancias indCLientes + 1, Qi
        '    Else
        '        'se aplica otra parte de la formula
        '    End If
        'End If



Public vector() As Integer
Public vectorA() As Integer

Sub Button2_Click()

    ReDim vector(5)
    ReDim vectorA(2, 5)

    Dim pruebando As Integer
    Dim funcion As Integer

    vector(2) = 3

    For ind = 1 To 5

        funcion = pruebaByRef(ind, pruebando)

        If funcion = 1 And pruebando = 0 Then
            MsgBox vector(ind)
        End If

        If funcion = 0 And pruebando = 1 Then
            MsgBox "Nada"
        End If
```

```
    Next ind

    MsgBox "Copio el vector"
    vectorA(1) = vector
    For ind = 1 To 5
        If vectorA(ind) <> Empty Then
            MsgBox vectorA(ind)
        Else
            MsgBox "Nada"
        End If
    Next ind


End Sub

Function pruebaByRef(ByVal ind As Integer, ByRef prueba As Integer) As Integer

    If vector(ind) <> Empty Then
        prueba = 0
        pruebaByRef = 1
    Else
        prueba = 1
        pruebaByRef = 0
    End If

End Function
```

ANEXO B. INSTANCIAS

Tabla 24.Datos de la instancia 1

P_i		$\overline{D}$	S_i	$r = 2$
1		1	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	52	2
2	13,0521561	11,3498337	57	2
3	16,7253639	6,62129582	59	4
4	8,38770714	10,1577807	67	3

Tabla 25. Datos de la instancia 2

P_i		$\overline{D}$	S_i	$r = 2$
2		1	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	52	2
2	-0,9514076	21,0328074	57	2
3	5,19809535	13,6658548	59	4
4	12,7301894	11,8188814	67	3

Tabla 26. Datos de la instancia 3

P_i		$\overline{D}$	S_i	$r = 2$
1		2	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	96	12
2	13,0521561	11,3498337	91	13
3	16,7253639	6,62129582	83	14
4	8,38770714	10,1577807	100	14

Tabla 27. Datos de la instancia 4

P_i		$\overline{D}$	S_i	$r = 2$
2		2	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	96	12
2	-0,9514076	21,0328074	91	13
3	5,19809535	13,6658548	83	14
4	12,7301894	11,8188814	100	14

Tabla 28. Datos de la instancia 5

P_i		$\bar{D}$	S_i	r = 2
1		1	2	
n	P_i		$\bar{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	52	12
2	13,0521561	11,3498337	57	13
3	16,7253639	6,62129582	59	14
4	8,38770714	10,1577807	67	14

Tabla 29. Datos de la instancia 6

P_i		$\bar{D}$	S_i	r = 2
1		2	1	
n	P_i		$\bar{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	96	2
2	13,0521561	11,3498337	91	2
3	16,7253639	6,62129582	83	4
4	8,38770714	10,1577807	100	3

Tabla 30. Datos de la instancia 7

P_i		$\overline{D}$	S_i	$r = 2$
2		1	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	52	12
2	-0,9514076	21,0328074	57	13
3	5,19809535	13,6658548	59	14
4	12,7301894	11,8188814	67	14

Tabla 31. Datos de la instancia 8

P_i		$\overline{D}$	S_i	$r = 2$
2		2	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	96	2
2	-0,9514076	21,0328074	91	2
3	5,19809535	13,6658548	83	4
4	12,7301894	11,8188814	100	3

Tabla 32. Datos de la instancia 9

P_i		$\overline{D}$	S_i	$r = 2$
1		1	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	52	2
2	13,0521561	11,3498337	57	2
3	16,7253639	6,62129582	59	4
4	8,38770714	10,1577807	67	3
5	9,33317057	11,2704855	64	2
6	18,3086642	14,926603	54	3
7	13,0393384	1,3104648	70	3
8	19,1271706	16,412244	59	3

Tabla 33. Datos de la instancia 10

P_i		$\overline{D}$	S_i	$r = 2$
2		1	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	52	2
2	-0,9514076	21,0328074	57	2
3	5,19809535	13,6658548	59	4
4	12,7301894	11,8188814	67	3
5	5,37230906	6,92064989	64	1
6	6,74818127	17,3638603	54	3
7	6,00586307	7,83782414	70	3
8	20,0325906	14,0157488	59	3

Tabla 34. Datos de la instancia 11

P_i		$\overline{D}$	S_i	$r = 2$
1		2	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	96	12
2	13,0521561	11,3498337	91	13
3	16,7253639	6,62129582	83	14
4	8,38770714	10,1577807	100	14
5	9,33317057	11,2704855	83	14
6	18,3086642	14,926603	97	12
7	13,0393384	1,3104648	84	15
8	19,1271706	16,412244	83	14

Tabla 35. Datos de la instancia 12

P_i		$\overline{D}$	S_i	$r = 4$
2		2	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	96	12
2	-0,9514076	21,0328074	91	13
3	5,19809535	13,6658548	83	14
4	12,7301894	11,8188814	100	14
5	5,37230906	6,92064989	83	14
6	6,74818127	17,3638603	97	12
7	6,00586307	7,83782414	84	15
8	20,0325906	14,0157488	83	14

Tabla 36. Datos de la instancia 13

P_i		$\overline{D}$	S_i	$r = 4$
1		1	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	52	12
2	13,0521561	11,3498337	57	13
3	16,7253639	6,62129582	59	14
4	8,38770714	10,1577807	67	14
5	9,33317057	11,2704855	64	14
6	18,3086642	14,926603	54	12
7	13,0393384	1,3104648	70	15
8	19,1271706	16,412244	59	14

Tabla 37. Datos de la instancia 14

P_i		$\overline{D}$	S_i	$r = 4$
1		2	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	96	2
2	13,0521561	11,3498337	91	2
3	16,7253639	6,62129582	83	4
4	8,38770714	10,1577807	100	3
5	9,33317057	11,2704855	83	2
6	18,3086642	14,926603	97	3
7	13,0393384	1,3104648	84	3
8	19,1271706	16,412244	83	3

Tabla 38. Datos de la instancia 15

P_i		$\overline{D}$	S_i	$r = 4$
2		1	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	52	12
2	-0,9514076	21,0328074	57	13
3	5,19809535	13,6658548	59	14
4	12,7301894	11,8188814	67	14
5	5,37230906	6,92064989	64	14
6	6,74818127	17,3638603	54	12
7	6,00586307	7,83782414	70	15
8	20,0325906	14,0157488	59	14

Tabla 39. Datos de la instancia 16

P_i		$\overline{D}$	S_i	$r = 4$
2		2	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	52	2
2	-0,9514076	21,0328074	57	2
3	5,19809535	13,6658548	59	4
4	12,7301894	11,8188814	67	3
5	5,37230906	6,92064989	64	2
6	6,74818127	17,3638603	54	3
7	6,00586307	7,83782414	70	3
8	20,0325906	14,0157488	59	3

Tabla 40. Datos de la instancia 17

P_i		$\overline{D}$	S_i	r = 2
1		1	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	52	2
2	13,0521561	11,3498337	57	2
3	16,7253639	6,62129582	59	4
4	8,38770714	10,1577807	67	3
5	9,33317057	11,2704855	64	2
6	18,3086642	14,926603	54	3
7	13,0393384	1,3104648	70	3
8	19,1271706	16,412244	59	3
9	6,48274178	18,0993072	53	2
10	4,94399854	7,79320658	65	4
11	13,6924345	8,85525071	56	2
12	6,18366039	11,5579699	67	4

Tabla 41. Datos de la instancia 18

P_i		$\overline{D}$	S_i	r = 2
2		1	1	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	52	2
2	-0,9514076	21,0328074	57	2
3	5,19809535	13,6658548	59	4
4	12,7301894	11,8188814	67	3
5	5,37230906	6,92064989	64	2
6	6,74818127	17,3638603	54	3
7	6,00586307	7,83782414	70	3
8	20,0325906	14,0157488	59	3
9	7,67368308	12,8104637	53	2
10	8,93145741	11,8996275	65	4
11	10,0300247	2,51754161	56	2
12	4,4709316	12,7626584	67	4

Tabla 42. Datos de la instancia 19

P_i		$\overline{D}$	S_i	r = 2
1		2	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	11,5726188	0,54506058	96	12
2	13,0521561	11,3498337	91	13
3	16,7253639	6,62129582	83	14
4	8,38770714	10,1577807	100	14
5	9,33317057	11,2704855	83	14
6	18,3086642	14,926603	97	12
7	13,0393384	1,3104648	84	15
8	19,1271706	16,412244	83	14
9	6,48274178	18,0993072	95	12
10	4,94399854	7,79320658	98	13
11	13,6924345	8,85525071	97	14
12	6,18366039	11,5579699	82	10

Tabla 43. Datos de la instancia 20

P_i		$\overline{D}$	S_i	r = 2
2		2	2	
n	P_i		$\overline{D}$	S_i
	Coord x	Coord y		
0	0	0	0	0
1	6,66598569	15,5276587	96	12
2	-0,9514076	21,0328074	91	13
3	5,19809535	13,6658548	83	14
4	12,7301894	11,8188814	100	14
5	5,37230906	6,92064989	83	14
6	6,74818127	17,3638603	97	12
7	6,00586307	7,83782414	84	15
8	20,0325906	14,0157488	83	14
9	7,67368308	12,8104637	95	12
10	8,93145741	11,8996275	98	13
11	10,0300247	2,51754161	97	14
12	4,4709316	12,7626584	82	10

Tabla 44. Datos de la instancia 21

P_i		$\overline{D}$	S_i	$r = 2$
1		1	2	
n	P_i Coord x	P_i Coord y	$\overline{D}$	S_i
0	0	0	0	0
1	11,5726188	0,54506058	52	12
2	13,0521561	11,3498337	57	13
3	16,7253639	6,62129582	59	14
4	8,38770714	10,1577807	67	14
5	9,33317057	11,2704855	64	14
6	18,3086642	14,926603	54	12
7	13,0393384	1,3104648	70	15
8	19,1271706	16,412244	59	14
9	6,48274178	18,0993072	53	12
10	4,94399854	7,79320658	65	13
11	13,6924345	8,85525071	56	14
12	6,18366039	11,5579699	67	10

Tabla 45. Datos de la instancia 22

P_i		$\overline{D}$	S_i	$r = 2$
1		2	1	
n	P_i Coord x	P_i Coord y	$\overline{D}$	S_i
0	0	0	0	0
1	11,5726188	0,54506058	96	2
2	13,0521561	11,3498337	91	2
3	16,7253639	6,62129582	83	4
4	8,38770714	10,1577807	100	3
5	9,33317057	11,2704855	83	2
6	18,3086642	14,926603	97	3
7	13,0393384	1,3104648	84	3
8	19,1271706	16,412244	83	3
9	6,48274178	18,0993072	95	2
10	4,94399854	7,79320658	98	4
11	13,6924345	8,85525071	97	2
12	6,18366039	11,5579699	82	4

Tabla 46. Datos de la instancia 23

P_i		$\overline{D}$	S_i	$r = 2$
2		1	2	
n	P_i Coord x	P_i Coord y	$\overline{D}$	S_i
0	0	0	0	0
1	6,66598569	15,5276587	52	12
2	-0,9514076	21,0328074	57	13
3	5,19809535	13,6658548	59	14
4	12,7301894	11,8188814	67	14
5	5,37230906	6,92064989	64	14
6	6,74818127	17,3638603	54	12
7	6,00586307	7,83782414	70	15
8	20,0325906	14,0157488	59	14
9	7,67368308	12,8104637	53	12
10	8,93145741	11,8996275	65	13
11	10,0300247	2,51754161	56	14
12	4,4709316	12,7626584	67	10

Tabla 47. Datos de la instancia 24

P_i		$\overline{D}$	S_i	$r = 2$
2		2	1	
n	P_i Coord x	P_i Coord y	$\overline{D}$	S_i
0	0	0	0	0
1	6,66598569	15,5276587	96	2
2	-0,9514076	21,0328074	91	2
3	5,19809535	13,6658548	83	4
4	12,7301894	11,8188814	100	3
5	5,37230906	6,92064989	83	2
6	6,74818127	17,3638603	97	3
7	6,00586307	7,83782414	84	3
8	20,0325906	14,0157488	83	3
9	7,67368308	12,8104637	95	2
10	8,93145741	11,8996275	98	4
11	10,0300247	2,51754161	97	2
12	4,4709316	12,7626584	82	4

ANEXO C. COSTO PROMEDIO DE CADA INSTANCIA

Tabla 48. Costo promedio instancia 1

INSTANCIA 1		
CORRIDA	COSTO	TIEMPO
1	97,90789	0
2	77,47274	0
3	103,0852	0
4	76,6436	0
5	90,7667	0
Promedio	**89,175226**	**0**

Tabla 49. Costo promedio instancia 2

INSTANCIA 2		
CORRIDA	COSTO	TIEMPO
1	93,351	0
2	128,7346	0
3	112,5899	0
4	115,4215	0
5	122,8602	0
Promedio	**114,59144**	**0**

Tabla 50. Costo promedio instancia 3

INSTANCIA 3		
CORRIDA	COSTO	TIEMPO
1	97,90789	0
2	77,47274	0
3	73,17127	0
4	98,78377	0,003906
5	104,9201	0,003906
Promedio	**90,451154**	**0,0015624**

Tabla 51. Costo promedio instancia 4

INSTANCIAS 4		
CORRIDA	COSTO	TIEMPO
1	123,6546	0,003906
2	147,9663	0,003906
3	117,7802	0,003906
4	88,72611	0
5	113,0378	0
Promedio	**118,233002**	**0,0023436**

Tabla 52. Costo promedio instancia 5

INSTANCIA 5		
CORRIDA	COSTO	TIEMPO
1	124,8255	0,003906
2	77,47274	0,003906
3	102,2094	0,003906
4	77,07958	0,003906
5	90,7667	0
Promedio	**94,470784**	**0,0031248**

Tabla 53. Costo promedio instancia 6

INSTANCIA 6		
CORRIDA	COSTO	TIEMPO
1	90,7667	0,003906
2	90,7667	0,003906
3	90,7667	0,003906
4	90,7667	0,003906
5	90,7667	0,003906
Promedio	**90,7667**	**0,003906**

Tabla 54. Costo promedio instancia 7

INSTANCIA 7		
CORRIDA	COSTO	TIEMPO
1	113,0378	0,003906
2	93,351	0
3	115,4215	0,003906
4	121,296	0
5	87,47657	0
Promedio	**106,116574**	**0,0015624**

Tabla 55. Costo promedio instancia 8

INSTANCIAS 8		
CORRIDA	COSTO	TIEMPO
1	112,5899	0,003906
2	128,7346	0,003906
3	115,4215	0,003906
4	93,351	0
5	120,0286	0
Promedio	**114,02512**	**0,0023436**

Tabla 56. Costo promedio instancia 9

INSTANCIA 9		
CORRIDA	COSTO	TIEMPO
1	172,7398	1,675781
2	146,9486	1,710938
3	195,6748	1,714844
4	148,8233	1,699219
5	173,2993	1,695313
Promedio	**167,49716**	**1,699219**

Tabla 57. Costo promedio instancia 10

INSTANCIA 10		
CORRIDA	COSTO	TIEMPO
1	162,4486	0,609375
2	190,7149	1,800781
3	211,2748	1,800781
4	190,1561	1,792969
5	162,2183	1,753906
Promedio	**183,36254**	**1,5515624**

Tabla 58. Costo promedio instancia 11

INSTANCIA 11		
CORRIDA	COSTO	TIEMPO
1	197,838	1,910156
2	182,5917	1,890625
3	189,0628	1,847656
4	193,3643	1,835938
5	170,8733	1,847656
Promedio	**186,74602**	**1,8664062**

Tabla 59. Costo promedio instancia 12

INSTANCIA 12		
CORRIDA	COSTO	TIEMPO
1	190,5742	1,882813
2	193,2883	1,882813
3	189,6585	1,851563
4	190,8161	1,894531
5	193,3677	1,871094
Promedio	**191,54096**	**1,8765628**

Tabla 60. Costo promedio instancia 13

INSTANCIA 13		
CORRIDA	COSTO	TIEMPO
1	175,2534	1,96875
2	145,8221	1,914963
3	183,4676	1,9375
4	162,1261	1,867188
5	147,6869	1,96875
Promedio	**162,87122**	**1,9314302**

Tabla 61. Costo promedio instancia 14

INSTANCIA 14		
CORRIDA	COSTO	TIEMPO
1	197,838	1,757813
2	193,5365	1,796875
3	170,5319	1,820313
4	191,8972	1,789063
5	198,4885	1,875
Promedio	**190,45842**	**1,8078128**

Tabla 62. Costo promedio instancia 15

INSTANCIA 15		
CORRIDA	COSTO	TIEMPO
1	186,5553	2,125
2	165,2685	2,117188
3	162,4319	2,117188
4	187,0032	2,132813
5	186,7855	2,1875
Promedio	**177,60888**	**2,1359378**

Tabla 63. Costo promedio instancia 16

INSTANCIA 16		
CORRIDA	COSTO	TIEMPO
1	190,5742	1,808594
2	192,4118	1,792969
3	190,4947	1,792969
4	208,0166	1,800781
5	201,7113	1,710938
Promedio	**196,64172**	**1,7812502**

Tabla 64. Costo promedio instancia 17

INSTANCIA 17		
CORRIDA	COSTO	TIEMPO
1	218,8571	1607,402
2	236,5996	1573,695
3	238,5175	1835,648
4	240,4079	1963,531
5	240,4079	1725,516
Promedio	**234,958**	**1741,1584**

Tabla 65. Costo promedio instancia 18

INSTANCIA 18		
CORRIDA	COSTO	TIEMPO
1	231,6838	1586,68
2	257,5828	1582,535
3	230,4667	1545,742
4	229,5043	1578,023
5	257,6696	1569,305
Promedio	**241,38144**	**1572,457**

Tabla 66. Costo promedio instancia 19

INSTANCIA 19		
CORRIDA	COSTO	TIEMPO
1	274,564	1893,391
2	230,4117	1898,305
3	231,8959	1785,617
4	267,0025	18,25121
5	230,6459	1873,289
Promedio	**246,904**	**1493,77064**

Tabla 67. Costo promedio instancia 20

INSTANCIA 20		
CORRIDA	COSTO	TIEMPO
1	235,3074	1890,043
2	247,7697	1882,699
3	255,3742	1893,027
4	254,0372	1888,004
5	224,8686	1859,297
Promedio	**243,47142**	**1882,614**

Tabla 68. Costo promedio instancia 21

INSTANCIA 21		
CORRIDA	COSTO	TIEMPO
1	246,6181	1821,586
2	278,3713	1847,871
3	259,0981	1827,738
4	220,0104	1878,918
5	244,0904	1903,098
Promedio	**249,63766**	**1855,8422**

Tabla 69. Costo promedio instancia 22

INSTANCIA 22		
CORRIDA	COSTO	TIEMPO
1	241,0092	1429,387
2	282,3015	1393,664
3	235,8133	1334,285
4	253,4322	1388,594
5	237,8119	1477,531
Promedio	**250,07362**	**1404,6922**

Tabla 70. Costo promedio instancia 23

INSTANCIA 23		
CORRIDA	COSTO	TIEMPO
1	249,5587	1830,391
2	237,2942	1775,148
3	252,1944	1739,82
4	231,1667	1699,191
5	247,7414	1800,047
Promedio	**243,59108**	**1768,9194**

Tabla 71. Costo promedio instancia 24

INSTANCIA 24		
CORRIDA	COSTO	TIEMPO
1	265,0628	1391,141
2	272,8732	1498,465
3	257,0734	1460,273
4	272,7236	1541,539
5	267,2904	1377,863
Promedio	**267,00468**	**1453,8562**

ANEXO D. ARTÍCULO PROYECTO

Un algoritmo Rollout (RA) para la solución del problema de ruteo de vehículos con demanda estocástica, desde una perspectiva de reoptimización.

Yesenia Katherine Núñez Núñez, Leslie Paola Robinson Quintero

Escuela de Estudios Industriales y Empresariales

Universidad Industrial de Santander

Bucaramanga, Colombia

yesenia.núñez@correo.uis.edu.co

leslie.robinson@correo.uis.edu.co

RESUMEN: En este artículo se presenta el Algoritmo Rollout de un solo paso (ORA) como una solución al Problema de Ruteo de Vehículo con Demanda Estocástica (VRPSD) con una perspectiva de reoptimización para el caso de un único vehículo, el cual se centra en hallar una política mejorada que minimice el costo de la ruta resultante. Esto se obtiene por mejoras realizadas secuencialmente y de manera cíclica, teniendo como base una solución a priori hallada a través de la heurística Nearest Neighbor (NN). Adicionalmente se construyó un aplicativo para generar un banco de pruebas con el objetivo de validar las soluciones obtenidas por el software y medir la calidad de los resultados, los cuales fueron comparados con el método evolutivo de colonia de hormigas (*Evolutionary Method based on Ant Colony Optimization,* EACO). Se realizó un diseño de experimentos con el objetivo de analizar los efectos de la variación de los parámetros del algoritmo Rollout sobre la variable respuesta. Utilizando la propuesta de investigación realizada por Silvia Galván [1] se ha construido un banco de pruebas, adaptando los factores y niveles al presente problema para validar el Algoritmo Rollout. Los resultados muestran que de las 24 instancias, el 37,5% indica que el Rollout obtiene mejores resultados que el método EACO.

PALABRAS CLAVE: VRP, VRPSD, Programación Dinámica, Algoritmo de Despliegue, Secuencia de Base.

ABSTRACT: This paper presents a one-step Rollout Algorithm (ORA) as a solution of Vehicle Routing Problem whit Stochastic Demands (VRPSD) with the prospect of reoptimization for an only-vehicle case, which is focused in finding out an improved policy that minimize the cost of the resulting route. This is gotten through the improvements which have been done sequentially and cyclically, taking as foundation an a-priori solution found through heuristic. Additionally a software application was made to create an answers bank with the objective of validating the solutions obtained by the software and taking measures of the quality of those results, which were compared using the *Evolutionary Method based on Ant Colony Optimization,*. Experiments were design to analyze the effect of variations of the parameters from the rollout algorithm on the variable answer. Using the investigation proposal made by Silvia Galvan, a test bank has been built, adapting the factors and levels to the current problem to ratify the Rollout algorithm. The results show that from 24 request, the 37,5% of them shows Rollout obtains better results than EACO.

KEYWORDS: VRP, VRPSD, Dynamic Programming, Algorithm Rollout, Base Sequence

1. INTRODUCCIÓN

En el ámbito de competencia que se desenvuelven hoy día las pequeñas, medianas y grandes empresas, el diseño de la red de distribución de un sistema logístico es de vital importancia para generar ventajas competitivas. El transporte es por excelencia uno de los procesos fundamentales de la estrategia logística de una organización, dado que suele ser el elemento individual con mayor ponderación en el consolidado de los costos logísticos de la mayoría de empresas. Por consiguiente, el diseño de rutas vehiculares es una manera de ofrecer un óptimo servicio y así reducir costos de transporte y satisfacer a los clientes en los tiempos de entrega.

Uno de los problemas de optimización combinatoria más populares, es el problema de ruteo de vehículos *(Vehicle Routing Problem, VRP),* su estudio ha generado técnicas de solución exactas y heurísticas de aplicación general. El VRP consiste en una serie de clientes que se encuentran geográficamente dispersos donde cada cliente presenta una demanda, se tiene un depósito y una flota de vehículos con capacidad limitada Q, el objetivo del problema es "encontrar un conjunto de rutas con costo mínimo visitando a cada uno de los clientes y satisfacer sus demandas sin violar las restricciones de la capacidad del vehículo"[2]. Para reflejar mejor la incertidumbre que existe en nuestra vida diaria, se asume un problema modelado por variables aleatorias tales como el conjunto de clientes a visitar, el tiempo de viaje entre los clientes y las demandas de los clientes, haciendo parte de unos parámetros no determinísticos, clasificándose en la literatura como problema de ruteo de vehículos estocástico *(Stochastic Vehicle Routing Problem, SVRP).* El presente trabajo se ocupa de una de las variantes del SVRP, en la cual se considera incertidumbre las demandas de los clientes, dado que esta es una variable aleatoria. Esta variante se conoce como Problema de Ruteo de Vehículos con Demandas Estocásticas (VRPSD).

El VRPSD se da cuando en un VRP las demandas de los clientes son variables estocásticas o aleatorias, éstas siguen una distribución de probabilidad conocida y las exigencias reales del cliente solo se conocen al llegar al sitio. Si se presenta una ruta "fallida" debido a que la demanda del cliente ha excedido la capacidad actual del vehículo o éste ha quedado sin existencias, la distancia recorrida se incrementa, debido naturalmente a que el vehículo se ve obligado a regresar al depósito. El objetivo del VRPSD es minimizar el costo total generado por la ruta prevista (a priori) y los viajes nuevos que se realizan hacia y desde el depósito. Para minimizar aún más los costos, Bertsimas et al. (1995) 3 diseña rutas a priori que puede determinar si regresa al depósito antes de que la capacidad del vehículo esté agotada, esta acción es denomina *Retorno Proactivo*. VRPSD ocurre tanto en situaciones de entrega y como en recogida. 4

Las principales investigaciones hechas sobre el VRPSD han estudiado tres enfoque principales que determina la política de servicio que abordará el vehículo para iniciar su ruta:

a) Enfoque de optimización a priori o política estática
b) Enfoque de reoptimización o dinámico
c) Enfoque mixto o política de descargue / abastecimiento

Como primera medida se encuentra el enfoque a priori el cual consiste en el diseño de rutas sin tener conocimiento de la demanda real del cliente, al momento del fracaso de la ruta, se adopta acciones de recursos sin necesidad de modificar el recorrido ya planificado. Se conoce también como política estática “de aquí y ahora”.

El enfoque de reoptimización, ocurre siempre y cuando se den actualizaciones del sistema con el servicio (múltiples estados), es decir, cada vez que el vehículo llegue a una ubicación y observa la demanda del cliente, el estado del sistema se actualiza. Esto da a entender que para resolver el VRPSD los datos no son conocidos completamente y esta información se revela a medida que pasa el tiempo, de manera dinámica, por consiguiente, se debe resolver el problema de manera secuencial. En la referencias [5] y [6] se desarrolla el Algoritmo Rollout (RA) para resolver el VRPSD con un solo vehículo bajo una aproximación dinámica, el cual fue el objeto de estudio del presente trabajo.

El tercer enfoque es mixto, esta política combina elementos de ambas políticas estáticas y dinámicas, donde el vehículo sigue una ruta a priori, pero también está habilitada con reglas dependientes de estado que permiten reabastecimientos anticipados[7]

El RA, propuesto originalmente por Bertsekas y Tsitsiklis (1996), Bertsekas et al. (1997), y Bertsekas (2000, 2001), supera la maldición de la dimensionalidad en la programación dinámica (DP) y el primer autor que proporciona una heurística computacional tratable es Secomandi en 1998 y Secomandi en 2001 8 , investiga un enfoque de reoptimización del VRPSD en el que se utiliza el RA para programación dinámica y así mejorar secuencialmente un recorrido a priori determinado, “en el que después de que es conocida la demanda de cada cliente, la parte restante del problema se vuelve a resolver por medio del uso de algoritmos de programación neuro-dinámica” 9 . Este enfoque genera mejores resultados que la estrategia de volver al depósito antes de que ocurra una acción que realmente lo obligue (estrategia preventiva de reposición de existencias) pero resulta ser mucho más complejo computacionalmente. La contribución de Secomandi en lo que se refiere a los métodos del RA es el desarrollo del Algoritmo Rollout de un solo paso (ORA).

El presente artículo está organizado de la siguiente manera. En la Sección 2 se describe el problema. En la Sección 3 se presenta una política Rollout para el VRPSD. En la Sección 4 se desarrolla el algoritmo Rollout. En la Sección 5 se presentan los resultados numéricos. En la sección 6 se muestra el diseño de experimentos realizado. Por último, en la Sección 7 se presentan las conclusiones de la investigación realizada.

1. DESCRIPCIÓN DEL PROBLEMA

1.1.Notación y supuestos

Existe un vehículo con capacidad fija Q, este parte del depósito para dirigirse hacia sus clientes, los cuales se encuentran ubicados en lugares distantes, entregando o recogiendo un bien. Entones, se denota que:

- $I = \{1, 2, \dots, n\}$, representa el conjunto de clientes
- $Nodo\, 0$, denota la ubicación del depósito
- $d\ i, j$, distancia entre clientes
- D_i, Variable aleatoria que describe la demanda del cliente, donde $i = 1,2, \dots, n$
- p_i, distribución de probabilidad en función de la demanda, siendo $$p_i\ r\ = P_r\ D_i = r\ , r = 0,1, \dots, R \leq Q$$

Los costos calculados son proporcionales a las distancias recorridas en el tour. La demanda de los clientes solo se conoce cuando el vehículo llega a su ubicación, como consecuencia, se pueden presentar *fallas de ruta* donde la demanda excede la capacidad del vehículo $D_i > Q$, por lo que el vehículo puede satisfacer parcialmente al cliente, ir al depósito para reabastecerse y seguir con la visita. Existe una política de ruteo donde el vehículo realiza una *ruta proactiva* en el cual se dirige al depósito a reabastecerse para recuperar su capacidad Q antes de que este quede sin existencias y ocurra un fallo de ruta. El objetivo de dar solución al problema es, encontrar una política de enrutamiento que minimice los costos y satisfacer completamente la demanda de todos los clientes.

1.2.Formulación del problema

Sea $X_k =\ l, q_l, r_1, \dots, r_n$ una matriz con $\mathrm{n} + 2$ componentes que representa el estado del sistema en la etapa de decisión k. Donde:

- l, ubicación actual del vehículo, donde $l \in 0,1., \dots, n$.
- q_l, capacidad actual del vehículo, donde $q_l\ \in 0,1, \dots, Q$
- r_i, demanda aún no entregada al cliente i, donde $i = 1, 2, \dots n$

r_i es "? ", si la demanda no se conoce y puede tomar un valor cualquiera $0, 1, \dots, R$

r_i es 0, si el cliente ha sido visitado y su demanda satisfecha.

r_i' es $1, 2, \dots, R - 1$, actualización de la demanda de un cliente cuando este ha sido visitado y parcialmente servido. Al tomar un valor conocido $p_r\ D_i = r'\ = 1$

El conjunto de estados (espacio) se denota como S. El número total de estados posibles en el sistema es elevado y es de orden $O\ nQR^n$. El sistema comienza en el estado $(0, Q, ?, ?, \dots, ?)$, donde el vehículo se encuentra en el nodo cero (depósito), con una capacidad total y demandas desconocidas. Terminando en el estado $0, Q, 0,0, \ldots ,0$, donde el vehículo vuelve al depósito recuperando la capacidad total y ha suplido las demandas de los clientes totalmente.

La variable aleatoria N representará el número de etapas desde el estado inicial hasta el estado final, su distribución de probabilidad se determina a partir de la distribución de las probabilidades de las demandas de los clientes por la ruta seguida. Un nuevo estado ocurre cuando el vehículo llega (no necesariamente por primera vez) a una ubicación del cliente.

La matriz $\pi = \mu_o, \mu_{1,}, \ldots, \mu_{N-1}$ es la política o secuencia de funciones a optimizar, siendo u_k un miembro del conjunto de controles o decisiones factibles para el estado x_k , esto es: $u_k \in U_k(x_k)$ y $U_k(x_k) = m \in 1, 2, \ldots, n \quad r_m \neq 0 \cup 0\} \times \{a{:}\, a \in \{0, 1\}\}$.

- u_k, representa el control tomado en la etapa k, dado por $\boldsymbol{u_k}\ \boldsymbol{m}, \boldsymbol{a}$ $\boldsymbol{m} \in 1, \ldots, n \ |r_m \neq 0 \cup 0$, es cualquier cliente próximo a visitar.

m toma el valor de 0 cuando las demandas son satisfechas y el sistema entra en estado de terminación.

$\boldsymbol{a} \in 0{,}1$, es la decisión de ir al depósito o no.

a toma el valor de 0 si el vehículo visita al cliente directamente y toma el valor de 1 si el vehículo se devuelve al depósito para luego continuar con el cliente.

En el caso de $a = 1$ se permiten viajes proactivos a la estación evitando fracasos de rutas y sea más costoso.

Si el sistema está en la etapa k y en el estado $X_k = l, q_l, r_1, \ldots, r_n \in S$ y se elige el control u_k entonces, el sistema pasa al estado $x_{k+1} = (m, q_m, r_1, \ldots, r'_m, \ldots, r_n)$ incurriendo en el costo de transición $g\ x_k, u_k$ con las siguientes probabilidades de transición $p\ x_k, x_{k+1}\ (u_k)$. El costo de transición $g(x_k, u_k)$ es proporcional a la distancia entre dos clientes, ya sea directamente o indirectamente (deteniéndose en el depósito).

La capacidad del vehículo después de servir al cliente m, q_m y la demanda del cliente m, r'_m, se actualiza de acuerdo con la realización de la demanda del cliente. El objetivo del problema es encontrar una política π que minimice el costo de ir J_N^π (*cost-to-go*) en la $N - etapa$, desde algún estado inicial. El costo óptimo de ir en la $N - etapa$ (es decir, costo esperado de terminación) para el estado x es $J_N^*\ x = min_{\pi \in}\ J_N^\pi(x)$ donde . es el conjunto de políticas admisibles

Si J_N^* es conocido por todos los estados, el control óptimo u_k^* para cada estado, resulta de la búsqueda del mínimo en la siguiente ecuación:

$$u_k^*\ x = \mu_k^*\ x = arg \min_{u_k \in U_k\ x_k} g\ x_k, u_k + \sum_{x_{k+1} \in S} p_{x_k x_{k+1}}\ u_k\ J_N^*\ x_{k+1} \ | \ x_k = x, \forall x \in S$$
(1)

La ecuación (1) muestra que en todas las etapas, las decisiones son clasificadas con base a la suma del precio actual y el precio futuro esperado, asumiendo una óptima toma de decisión para las etapas subsiguientes. Desde esta sección se simplificará el subíndice N en la relación del costo de ir para simplificar la notación.

Bajo el algoritmo Rollout de una sola etapa (ORA), un control aproximado u_k para un estado particular x_k alcanzado en la etapa k resulta de aproximar el costo de ir optimo $J^* \ x_{k+1}$ en la ecuación (1) por $J(x_{k+1})$. Esta aproximación es hecha para todos los estados $x_{k+1} \in S$ que puedan ser alcanzados un paso hacia delante de la etapa actual k. Este valor óptimo es aproximado debido a lo complicado que es obtener dicho costo.

Dejando $J(x_k)$ ser el costo de ir de la política de base en el estado x_k , se toma una decisión de acuerdo con la siguiente minimización:

$$u \ x_k$$

$$= \arg \min_{u\in U(x_k)} \quad g(x_k, u_k)$$

$$+ \underset{x_{k+1}\in S}{\quad} P_{x_k x_{k+1}} \ u_k \ .J(x_{k+1}) \qquad (2)$$

Si la evaluación exacta de $J(x_{k+1})$ está disponible, entonces se puede fácilmente demostrar que la política de despliegue resultante mejora la calidad de la política común (Bertsekas y Tsitsiklis 1996). Tenga en cuenta que la aplicación eficaz de este método requiere que $J(x_{k+1})$ sea calculada eficientemente.

2.UNA POLITICA ROLLOUT PARA EL VRPSD

La mayor dificultad en la solución del VRP es debido a la cantidad de rutas posibles existentes en el problema, debido al número de clientes involucrados, por lo que al aumentar el número de clientes el problema se hará más complejo y representará un mayor uso de recursos como el tiempo y espacio (cantidad de memoria). Entonces, la cantidad de rutas que existen con un número n de clientes sería una permutación $n!$. Cuando se analiza el VRPSD, el dilema no solo radica en todas las posibles rutas, ahora se suma el hecho de que los clientes poseen demandas estocásticas y esto hace que incremente la dificultad del ejercicio. Por este y otros motivos se ha buscado a lo largo de la historia métodos que ayuden a la solución de estos problemas de manera más eficiente. El algoritmo Rollout representa una de las respuestas a la búsqueda exhaustiva de procedimientos para la solución del VRPSD de una manera aproximada y la optimización de recursos.

El Rollout parte desde la consideración de una secuencia de base o tour base τ disponible para el problema y calcula el costo de ir esperado $J(x_{k+1)}$ eficientemente. Seleccionando el tour a priori de manera correcta, el Rollout arroja una política de mejora τ. Para la aplicación de este algoritmo, se emplea una heurística cíclica ζ eficiente y consistente, Bertsekas y Tsitsiklis en 1996 y Bertsekas et al. en 1973 afirman que si el tour base es calculado con la heurística cíclica, la política resultante es mejor que la política inicial.

2.1.Secuencia de base o política a priori

El primer paso para la implementación del Rollout es encontrar una secuencia de base por medio de la heurística Nearest Neighbor (NN) para el TSP, definido como:

Dado un conjunto de puntos $P = \{p_1, \ldots, p_n\}$ en un espacio vectorial X, se da un nuevo punto de consulta $q \in X$, encontrar en el conjunto P el punto que este más cercano a q eficientemente. Teniendo en cuenta que para la construcción de una ruta, el vehículo comienza y termina en el depósito (denotado por el nodo 0), la definición del NN al algoritmo Rollout sería, dado un conjunto de clientes $I = \{1,2, \ldots, n\}$, se debe encontrar el cliente más próximo al depósito, ignorando las demandas de los clientes y aplicando el vecino más cercano. Una vez elegido el cliente más cercano al depósito, a partir de allí se debe repetir el procedimiento para los clientes faltantes hasta completar una ruta llamada secuencia de base (SB), tour base o política a priori denotada por τ.

Se considera la secuencia de base $\tau = (0, 1, 2, \ldots n, 0)$ donde el vehículo visita a los clientes siguiendo esa ruta, se tiene en cuenta los retornos al depósito en caso de un fallo de ruta o al darse reposiciones tempranas durante el recorrido. Al seguir esta ruta, se realizan actualizaciones de la secuencia de base que ayudan a recalcular el costo de ir aproximado.

2.2.Cálculo de los controles en la política Rollout.

El cálculo de los controles, a lo largo del recorrido, son necesarios para tomar decisiones y discernir hacia qué cliente debe ir el vehículo y cómo ir hasta él.

Se parte de la ruta a-priori obtenida por el Nearest Neighbor (NN) y el vehículo visitas a los clientes en ese orden dado, y tiene que elegir, de acuerdo a la demanda del cliente actual, si se procede al siguiente cliente o si antes de ir al cliente va al depósito para un reabastecimiento.

Considerando la definición del VRPSD, supóngase que el servicio en el cliente actual j ha terminado y el vehículo tiene una carga q restante y siendo $f_j(q)$ el costo total esperado del nodo j en adelante. Con esta notación, el costo esperado de la ruta a-priori es $f_0(Q)$. Si L_j representa el conjunto de todas las cargas posibles que un vehículo puede tener después de la terminación del servicio en el cliente j, entonces, $f_j(q)$ para $q \in L_j$ satisface:

$$\boldsymbol{f_j(q) = \min\{f_j^p(q), f_j^r(q)\}}, \quad (3)$$

donde

$$f_j^p(q) = d_{j,j+1} + \sum_{k:k\leq q} f_{j+1}(q-k)p_{j+1,k} + \sum_{k:k>q} \left[2d_{j+1,0} + f_{j+1}(q+Q-k)\right] p_{j+1,k} \quad (4)$$

$$f_j^r(q) = d_{j,0} + d_{0,j+1} + \sum_{k=1}^{K} f_{j+1}(Q-k)p_{j+1,k} \quad (5)$$

Con la condición límite:

$$f_n(q) = d_{n,0}, q \in L_n \quad (6)$$

Siendo:

- $f_j^p(q)$, costo esperado de ir directamente al cliente
- $f_j^r(q)$, costo esperado de ir al cliente haciendo primero un retorno al depósito
- d_{ij}, distancia entre nodos

- $p_{j+1,k}$, Probabilidad de que la demanda en el nodo $j+1$ sea igual a k
- k, demanda del cliente
- q, capacidad residual del vehículo

Se debe tener en cuenta que en la fórmula (4) si $k \leq q$ se aplica $d_{j,j+1} + \quad_{k:k\leq q} f_{j+1}(q-k)p_{j+1,k}$, y/o si $k > q$ se aplica $d_{j,j+1} + \quad_{k:k>q} 2d_{j+1,0} + f_{j+1}(q+Q-k)\ p_{j+1,k}$. La fórmula (6) es el resultado de ir desde el último cliente por visitar hasta el depósito, valor que será remplazado de atrás hacia adelante en cada función indicada f_{j+1} de cada etapa.

3.DESARROLLO DEL ALGORITMO ROLLOUT

Se tiene una secuencia de base $\tau = (0, 1, 2, \ldots n, 0)$ y se asume que el vehículo visita a los clientes de acuerdo con τ, retornando al depósito en caso en que hayan rutas fallidas o si se realiza un reabastecimiento preventivo. Bertsimas (1992) propone una heurística cíclica ζ que realiza actualizaciones en la secuencia de base cambiando el orden de dicha secuencia (en un solo paso ORA) y construyendo una ruta cíclica o secuencia de base actualizada (*Updated Base Sequence UBS*) τ_t^{ζ} que comienza en la locación t y sigue a τ cíclicamente, omitiendo los clientes que se encuentran completamente servidos.

$$\tau_t^{\zeta} = (t, t+1, \ldots, n, 1, \ldots, t-1, 0)$$
(7)

La distancia esperada o el costo de ir aproximado de τ_t^{ζ} en el estado x_k es $J(x_{k+1})$, el cual hace parte de la ecuación (2) descrita en la formulación del problema, este costo se asocia a la distancia y se halla usando las ecuaciones (4), (5) y (6).

Partiendo del depósito el vehículo se dirige al primer cliente, siguiendo la secuencia de base a priori τ, tomando siempre un control de $u_k(m, 0)$ para la primera etapa, la demandan del cliente t es observada y completamente servida (porque $k \leq Q$) y la capacidad del vehículo se actualiza. El sistema se mueve a un estado x_{k+1} y un control debe ser calculado para seguir el recorrido y decidir hacia qué cliente ir. En cada estado x_k el vehículo debe tomar un control u_k en donde después de haber servido al cliente actual t, se origina una secuencia de base actualizada τ_t^{ζ}.

Hallando la distancia esperada para cada secuencia de base actualizada, se procede a escoger el de menor valor entre $\{f_j^p\ q$, $f_j^r\ q\ \}$ y este valor es comparado junto con las otras secuencias de base actualizadas, donde el mínimo en esa etapa dará a conocer el control que se deberá tomar para pasar al estado x_{k+1} (siguiendo τ). Este procedimiento se realiza a lo largo de la ruta a-priori y luego sobre la actualización de base seleccionada, hasta que todos los clientes han sido totalmente servidos, el vehículo retorna al depósito y resulta una ruta final mejorada τ junto con un conjunto de controles admisibles mejorados π.

La distancia total esperada de la ruta final, teniendo en cuenta cada control u_k, se halla de la siguiente manera:

- Si en el estado x_k un control $\boldsymbol{u_k}\ \boldsymbol{m}, \mathbf{0}$ fue tomado, entonces la distancia que será computada en esta etapa será $\boldsymbol{d}(\boldsymbol{l}, \boldsymbol{m})$

- Si en el estado x_k un control $\boldsymbol{u_k}\ \boldsymbol{m}, \mathbf{1}$ fue tomado, entonces la distancia que será computada en esta etapa será $\boldsymbol{d}\ \boldsymbol{l}, \mathbf{0}\ \ + \boldsymbol{d}(\mathbf{0}, \boldsymbol{m})$

4.RESULTADOS NUMERICOS

4.1.Generación de instancias

Una instancia comprende la asignación de los valores a un conjunto de parámetros característicos del problema. Por medio de las instancias se analizan los resultados obtenidos por el método de solución propuesto.

Los parámetros para el VRPSD tenidos en cuenta en la generación de las instancias son:

- Capacidad del vehículo (Q)
- Número de clientes
- Demanda promedio de cada cliente
- Probabilidad de las demanda
- Ubicación de los clientes

En el caso de problemas clásicos como el VRP, existen diversas fuentes de instancias o conjuntos de datos en la literatura, sin embargo, Bianchi *et al.* señalan que comúnmente no se utiliza el Benchmark para el VRPSD. Por tanto, varios autores han desarrollado sus propios bancos de pruebas, por ejemplo "Yang *et al*. y Teodorovic y Lucic, generaron aleatoriamente la ubicación del depósito y de los clientes, la demanda de cada cliente, la media y la varianza."[11]

Utilizando la propuesta de investigación realizada por Silvia Galván se ha construido un banco de pruebas, adaptando los factores y niveles al presente problema para validar el Algoritmo Rollout.

Para la generación de las instancias se determinan dos niveles para cada uno de los factores. El número de clientes toma los valores de $n = [4, 8, 12]$ para los niveles 1 y 2. La ubicación de los clientes en el nivel 1 presenta una distribución uniforme dentro del intervalo [0, 20] y en el nivel 2 se presenta una distribución normal. La desviación tomada, con respecto a la demanda, está comprendida en el intervalo de [1, 5] para el nivel 1 y [80, 100] para el nivel 2. Por tal razón, las cuatro posibles demandas de cada cliente toma valores en el intervalo de $[D_i - S_i, D_i + S_i]$ y la probabilidad de ocurrencia para cada una de las posibles demandas de los clientes presenta una distribución de probabilidad uniforme.

La capacidad del vehículo se calcula de acuerdo a la siguiente fórmula [12]:

$$Q = \frac{{}_{i=1}^{n}(D_i) * r}{n}$$

Donde:

r es el número promedio de clientes atendidos por el vehículo antes de ir al depósito. Debido a que el número máximo de clientes a introducir en el programa es de 12, se realizó para cada grupo de clientes una prueba en la cual se varia el r para observar el valor más adecuado a utilizar dentro del problema Rollout, teniendo en cuenta que la capacidad del vehículo debe ser mayor a la demanda máxima que posean los clientes y que los controles resultantes reflejen las idas al depósito, ya que con una capacidad muy grande para el problema el vehículo cumplirá la ruta sin necesidad de hacer retornos al depósito. A continuación en la tabla 4 se muestra la capacidad del vehículo para cada grupo de clientes:

Número de clientes	Número de clientes atendidos antes de ir al depósito		
	$r = 2$	$r = 4$	$r = 8$
4	117	235	470
8	121	241	482
12	121	241	482

Tabla 1. Capacidad del vehículo promedio para cada grupo de clientes variando r

De acuerdo a lo anterior, se escoge el valor de $r = 2$, ya que con este valor se observa una capacidad del vehículo conveniente para cada grupo de clientes y poder analizar la interacción que existe entre factores y niveles dentro del problema.

D_i representa la demanda promedio para cada cliente i, el valor de r es el número promedio de clientes atendidos por el vehículo antes de ir al depósito. En este caso toma el valor de 2 los dos niveles. Se debe tener en cuenta que la capacidad del vehículo siempre debe ser mayor o igual a la máxima demanda de los clientes.

A continuación se presenta la tabla 5 de factores y niveles:

	Nivel 1	Nivel 2
Número de clientes (n)	4, 8, 12	4, 8, 12
Ubicación Clientes (p)	[0, 20] Distribución uniforme	(10, 5) Distribución normal
Demanda promedio (D_i	[50, 70]	[80, 100]
Desviación (S_i)	[1, 5]	[10, 15]

Tabla 2. Factores y niveles

El banco de pruebas está conformado por ocho instancias para cada conjunto de clientes $n = [4, 8, 12]$, entonces, en total existen veinticuatro instancias generadas y cada una contiene una combinación de factores y niveles. A continuación se presenta las tablas 6, 7 y 8 con la información de las instancias para cada conjunto de clientes.

Instancias	Número de clientes	Ubicación de los clientes	Demanda promedio	Desviación
1	4	Nivel 1	Nivel 1	Nivel 1
2	4	Nivel 2	nivel 1	Nivel 1
3	4	Nivel 1	Nivel 2	Nivel 2
4	4	Nivel 2	Nivel 2	Nivel 2
5	4	Nivel 1	Nivel 1	Nivel 2
6	4	Nivel 1	Nivel 2	Nivel 1
7	4	Nivel 2	Nivel 1	Nivel 2
8	4	Nivel 2	Nivel 2	Nivel 1

Tabla 3. Banco de pruebas para 4 clientes

Instancias	Número de clientes	Ubicación de los clientes	Demanda promedio	Desviación
9	8	Nivel 1	Nivel 1	Nivel 1
10	8	Nivel 2	Nivel 1	Nivel 1
11	8	Nivel 1	Nivel 2	Nivel 2
12	8	Nivel 2	Nivel 2	Nivel 2
13	8	Nivel 1	Nivel 1	Nivel 2
14	8	Nivel 1	Nivel 2	Nivel 1
15	8	Nivel 2	Nivel 1	Nivel 2
16	8	Nivel 2	Nivel 2	Nivel 1

Tabla 4. Banco de pruebas para 8 clientes

Instancias	Número de clientes	Ubicación de los clientes	Demanda promedio	Desviación
17	12	Nivel 1	Nivel 1	Nivel 1
18	12	Nivel 2	nivel 1	Nivel 1
19	12	Nivel 1	Nivel 2	Nivel 2
20	12	Nivel 2	Nivel 2	Nivel 2
21	12	Nivel 1	Nivel 1	Nivel 2
22	12	Nivel 1	Nivel 2	Nivel 1
23	12	Nivel 2	Nivel 1	Nivel 2
24	12	Nivel 2	Nivel 2	Nivel 1

Tabla 5. Banco de pruebas para 12 clientes

4.2.Factores del diseño de experimentos

Para el análisis de los factores que influyen en la respuesta de la función objetivo, se eligió el diseño factorial 2^3, en donde cada factor se estudia a dos niveles y se contemplan todas las combinaciones de cada nivel de un factor con todos los niveles de los otros factores. En este caso se consideran tres factores: ubicación de los clientes, demanda promedio, desviación y dos niveles para cada uno de ellos: nivel 1 y nivel 2. En este caso el algoritmo no tiene parámetros definidos, por tanto, para el problema del Rollout se trabaja con tres factores del VRPSD. El número de réplicas para cada instancia es de 4. A continuación se muestra los 8 tratamientos que se realizarán para cada grupo de clientes.

	A Ubicación	B Demanda	C Desviación
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

Tabla 6. Tratamientos del diseño factorial 2^3

5.DISEÑO DE EXPERIMENTO

5.1.Resultado del diseño experimental

El algoritmo Rollout fue programado en Excel 2010 y ejecutado en un equipo procesador Intel Core i5 con 6GB de RAM. A continuación se muestra los resultados obtenidos de las instancias al ser ejecutados en el software para validar la información obtenida. Cada instancia tiene 5 réplicas las cuales se han hecho para tomar un costo y tiempo promedio. Las tablas 10, 11 y 12 muestran los resultados del diseño de experimento para cada grupo de clientes.

Instancias	Corridas			Valor promedio de la función objetivo para 4 clientes
	A	B	C	
Instancia 1	-	-	-	89,175526
Instancia 2	+	-	-	114,59144
Instancia 3	-	+	-	90,7667
Instancia 4	+	+	-	114,02512
Instancia 5	-	-	+	94,470784
Instancia 6	+	-	+	106,116574
Instancia 7	-	+	+	90,451154
Instancia 8	+	+	+	118,233002

Tabla 7. Resultado del diseño experimental para 4 clientes.

Instancias	Corridas			Valor promedio de la función objetivo para 8 clientes
	A	B	C	
Instancia 9	-	-	-	167,49716
Instancia 10	+	-	-	183,36254
Instancia 11	-	+	-	190,45842
Instancia 12	+	+	-	196,64172
Instancia 13	-	-	+	162,87122
Instancia 14	+	-	+	177,60888
Instancia 15	-	+	+	186,74602
Instancia 16	+	+	+	191,54096

Tabla 8. Resultado del diseño experimental para 8 clientes.

Instancias	Corridas			Valor promedio de la función objetivo para 12 clientes
	A	B	C	
Instancia 17	-	-	-	234,958
Instancia 18	+	-	-	241,38144
Instancia 19	-	+	-	250,07362
Instancia 20	+	+	-	267,00408
Instancia 21	-	-	+	249,63766
Instancia 22	+	-	+	243,59108
Instancia 23	-	+	+	246,904
Instancia 24	+	+	+	243,47142

Tabla 9. Resultado del diseño experimental para 12 clientes.

5.2.Análisis del diseño experimental

A cada grupo de clientes del banco de pruebas se le aplicó un análisis de diseño de experimento seleccionado. A continuación se presenta los efectos estimados de cada uno de los factores para cada grupo sobre la variable respuesta.

Factores	Efecto estimado para 4 clientes
A: Ubicación clientes	23,8663
B: Demanda promedio	1,04194
C: Desviación	-4,95912

Tabla 10. Efectos estimados para la variable respuesta para 4 clientes.

Factores	Efecto estimado para 8 clientes
A: Ubicación clientes	13,3646
B: Demanda promedio	26,5669
C: Desviación	-2,90337

Tabla 11. Efectos estimados para la variable respuesta para 4 clientes.

Factores	Efecto estimado para 12 clientes
A: Ubicación clientes	0,26692
B: Demanda promedio	7,31942
C: Desviación	-4,5636

Tabla 12. Efectos estimados para la variable respuesta para 4 clientes.

El signo positivo de un efecto estimado indica que al pasar de un nivel bajo a un nivel alto el costo de la ruta aumenta, esto se refleja en los factores A y B de cada grupo de clientes, mientras el factor C tiene un valor negativo. El objetivo del problema es minimizar el costo de la ruta recorrida, por tanto, la ubicación de los clientes y la demanda promedio deben optar un nivel bajo y la desviación tomar un nivel alto.

CONCLUSIONES

Los objetivos planteados para esta investigación se cumplieron en su totalidad. El algoritmo Rollout se desarrolló en Microsoft Excel 2010 para resolver el problema de ruteo de vehículos con demandas estocásticas bajo una perspectiva de reoptimización, arrojando resultados factibles.

A partir de la revisión bibliográfica, se decide tratar el algoritmo Rollout de un solo paso (*one-step rollout algorithm ORA*), el cual es un tipo de regla de iteración de un paso de mejoramiento sobre los estados del sistema, con el objetivo de encontrar una política mejorada π.

El Rollout es un algoritmo que parte de una ruta a priori y empleando una heurística cíclica tiene como finalidad optimizar la toma de decisiones (controles) en cada etapa, para finalmente obtener una aproximación al costo óptimo de la ruta resultante.

A partir del resultado del diseño de experimentos se concluye que:

- El análisis del algoritmo se dividió para 4, 8 y 12 clientes. En los efectos estimados, para 4 y 8 clientes la ubicación de los clientes y la demanda promedio respectivamente, son factores significativos sobre la variable de respuesta.

- Para obtener mejores resultados sobre la función objetivo la ubicación de los clientes y la demanda promedio deben tomar el nivel 1 mientras que la desviación debe tomar el nivel 2.

- En el grupo de 12 clientes ningún factor es significativo sobre la variable respuesta.

- El análisis de varianza muestra que las interacciones entre los factores no tiene influencia en ningún grupo de clientes sobre el costo de la ruta final.

- Se trabaja con parámetros propios del problema de ruteo de vehículo con demandas estocásticas, siendo tres de estos los factores analizados para el algoritmo Rollout.

En cuanto a los controles o decisiones tomadas por el vehículo a lo largo de la trayectoria, se observa que el programa arroja resultados coherentes, realizando los retornos necesarios al depósito. Teniendo en cuenta la aleatoriedad de las demandas de cada cliente, en cada corrida se obtiene controles diferentes y como consecuencia un costo distinto.

Los resultados obtenidos al comparar el Rollout con el método EACO muestran que el 37.5% de las instancias indican que el algoritmo Rollout arroja mejores resultados para el grupo de 8 y 12 clientes, con una diferencia porcentual mayor al 1%.

REFERENCIAS

[1] GALVÁN, Silvia; ARIAS, Javier y LAMOS, Henry. En: DYNA. Junio, 2013. Ed,179. pp. 60-69

[2] NOVOA, Clara., et al. A Set-Partitioning-Based Model for the Stochastic Vehicle Routing Problem. En: Computational Optimization Research At Lehigh:Technical Report. [en línea]. [consultado 12 junio 2013]. (2006); Disponible en: <http://www.optimizationonline.org/DB_FILE/2006/12/1542.pdf>

[3] BERTSIMAS, Dimitris; CHERVI, Philippe Y PETERSON, Michael. Computational Approaches to Stochastic Vehicle Routing Problems. En: Transportation Science. Vol. 29, No. 4 (1995); p. 342–352.

[4] NOVOA, Clara. The Vehicle Routing Problem with Stochastic Demands Combining Static and Dynamic Approaches. p 1-7.

[5] NOVOA, Clara y STORER, Robert. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. En: European journal of operational research. Vol. 196, No 1 (2009); p 509-515.

[6] SECOMANDI, Nicola. A rollout policy for the vehicle routing problem with stochastic demands. En: Institute for Operation Research and de Management Science. Vol. 49, No. 5, (2001); p 798.

[7] SECOMANDI, Nicola. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. En: Computers & Operations Research. Vol. 27. (2000); p. 1205.

[8] BIANCHI, Leonora, et al. Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Journal of Mathematical Modelling and Algorithms. (2006); p. 91-110.

[9] TATARAKIS, A y MINIS, I. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. En: European Journal of Operational Research. Vol. 34, (2009); p. 558

[10] NOVOA, Clara y STORER, Robert. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. En: European journal of operational research. Vol. 196, No 1 (2009); p 509-515.

[11] TAN, K.C; CHEONG, C.Y y GOH C.K.. En: European Journal of Operational Research. Vol. 177, (2007); p. 813-839.

[12] GENDREAU, M., LAPORTE, G. y SÉGUIN, R., An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers. En: Transportation Science.1995.pp. 143-155.