

Método de Agrupamiento no Supervisado para el Problema de Ruteo Vehicular con Restricciones de Capacidad en Vehículos

Alina Martínez-Oropeza¹, Marco Antonio Cruz-Chávez²

¹Posgrado en Ingeniería y Ciencias Aplicadas,

²Centro de Investigaciones en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del Estado de Morelos. Av. Universidad 1001, Col. Chamilpa, 62209, Cuernavaca, Morelos, MÉXICO

{mcruz, alinam}@uaem.mx

Resumen. En este artículo se propone un método de agrupamiento no supervisado (CA-CVRP) para el problema de Ruteo Vehicular con Restricciones de Capacidad en Vehículos y se presentan las ventajas y desventajas de algunos de los algoritmos de agrupamiento más utilizados para tratar problemas de Ruteo Vehicular.

Se realizaron pruebas experimentales utilizando benchmarks de 100 y 1000 clientes con diferentes tipos de distribución, con lo que se demostró que el algoritmo propuesto es eficaz y rápido, mostrando mejoras considerables a algunas de las desventajas presentadas por algoritmos de agrupamiento existentes para el problema tratado.

Palabras Clave: Centroide, Cliente, Distancia Euclidiana, Población Heterogénea, k-Means

1 Introducción

En la actualidad existen gran cantidad de problemas de optimización considerados difíciles de resolver, para los cuales se han implementado variedad de métodos heurísticos tratando de obtener mejoras a las cotas conocidas. De acuerdo a la literatura, se ha observado que el realizar una segmentación adecuada del problema permite abordarlos de una manera más sencilla mediante algún método de optimización. Para llevar a cabo dicha segmentación, algunos de los métodos más utilizados son los conocidos como procesos no supervisados, enfocados a la generación de agrupamientos [8, 13], los cuales son procedimientos de segmentación que permiten encontrar similitudes y diferencias entre datos pertenecientes a una población heterogénea y generar grupos disjuntos de elementos.

El análisis llevado a cabo por los métodos de no supervisados basados en agrupamientos, mejor conocidos como algoritmos de agrupamiento, dividen una

población de datos heterogéneos en grupos, manteniendo la distribución natural de los datos. En algunos casos, el resultado obtenido a partir de estos algoritmos es únicamente una solución inicial para la aplicación de otro método de solución. Esta es una de las razones principales por lo que estos métodos juegan un papel importante en diferentes áreas de la investigación, tales como [8] psicología y ciencias sociales, biología, estadística, reconocimiento de patrones [10], compresión de imágenes y video y más recientemente en minería de datos [5], donde se han aplicado a problemas de optimización considerados difíciles de resolver.

Los algoritmos de agrupamiento han sido ampliamente utilizados, en el área de minería de datos aplicándolos a diversos problemas de optimización clasificados como NP-Complejos, tales como el problema de Ruteo Vehicular y sus variantes [1, 3], Coloreo de Grafos [2], entre otros.

En la actualidad gran cantidad de trabajos de investigación abordan diversos problemas clasificados como difíciles de resolver, mediante la aplicación de métodos de agrupamiento. Algunos de los trabajos que fueron relevantes para este trabajo de investigación son los presentados en [25], donde se proponen tres algoritmos de agrupamiento, los cuales son aplicados para construir un selector de algoritmos, en base a una muestra de 2430 casos obtenidos a partir de 7 diferentes heurísticas aplicados al Problema de Bin Packing. Para cada caso tratado se utiliza la técnica de aprendizaje C4.5, la cual permite aprender patrones de agrupamiento de casos de Bin Packing. Esta técnica involucra tres métodos denominados *CIGIC* (*Clustering by Initial Grouping, Integration and Validation*), el cual realiza en primera instancia un *agrupamiento inicial* basado en las regiones simples del problema, posteriormente se realiza una *fase de integración*, donde los casos de las regiones múltiples se incorporan a uno de los grupos definidos por las regiones simples, para finalmente pasar a la *fase de validación* para confirmar que los casos de las regiones múltiples fueron integrados al grupo con mayor afinidad, lo cual se lleva a cabo aplicando el algoritmo k-medias. *CIG* (*Clustering by Initial Grouping*), éste método considera que cada conjunto de algoritmos seleccionado para cada caso forma un agrupamiento, por lo que para la agrupación inicial asigna etiquetas a cada uno de los casos. A diferencia del método anterior, cada grupo obtenido cuenta con un conjunto de casos y un conjunto de los mejores algoritmos para dichos casos, y *CIGP* (*Clustering by Initial Grouping and Partitioning*), el cual realiza un agrupamiento inicial para posteriormente dividir cada grupo en dos subgrupos aplicando k-medias y tomando en cuenta como atributos de agrupamiento los indicadores de complejidad, conservando aquellos grupos donde el caso más lejano y el más cercano al centro del grupo rebasa un umbral definido en el método. De acuerdo a las pruebas experimentales, el método CIG obtuvo los mejores resultados para los casos de Internet, mientras que para la fase de entrenamiento, los métodos CIGI y CIGIV fueron la mejor opción.

En [26] se propone el algoritmo CCA (*Capacited Clustering Algorithm*), el cual es una técnica de descomposición aplicada al tratamiento del Problema General de Distribución (GDP). El algoritmo propuesto consiste en descomponer el problema en subproblemas más pequeños, mediante la aplicación del método k-means clásico, generando un número de agrupamientos definido, basados en la distancia, además de involucrar la generación de un centroide extra (gran centroide), el cual se encarga de ordenar y asignar nodos a grupos nuevos tomando en cuenta la capacidad del vehículo. Cabe mencionar que durante esta fase, los todos centroides son

recalculados, hasta que no existan elementos aislados. Este método fue aplicado a un algoritmo de 3 fases, donde la primera corresponde al algoritmo CCA, la segunda aplica un algoritmo de Recocido Simulado y la tercera fase corresponde a un método heurístico de mejora, donde se aplica una estructura 3-opt. El método propuesto fue probado con una instancia real de 10 vehículos con la misma capacidad (500 unidades), 110 bomberos y 5 depósitos en diferentes regiones. Los resultados fueron comparados con los obtenidos por 6 heurísticas: *Old Strategy*, *Routing First and Cluster Second* (RFCS), RFCS con una estructura 3-opt, RFCS con Recocido Simulado, *Cluster First and Routing Second* (CFRS) y CFRS con una estructura 3-opt, mejorando los resultados obtenidos por las heurísticas antes mencionadas.

En [27] se presenta un algoritmo de agrupamiento k-means mejorado aplicado al problema de Ruteo Vehicular con Ventanas de Tiempo (VRPTW), el cual consta de 4 fases; la primera corresponde a la selección (inicialización) de los puntos de inicio o centroides, la segunda fase corresponde a la clasificación, asignación o distribución del conjunto de clientes en grupos, utilizando la regla del cliente o centroide más cercano, calcula la similaridad utilizando el error al cuadrado y el total del cuadrado del error con la finalidad de verificar la variación existente dentro del agrupamiento, calcula la distancia entre los clientes utilizando la fórmula de la distancia euclidiana, con la finalidad de generar grupos que minimicen la distancia en su interior; la tercera fase corresponde a la fase de actualización, la cual se encarga de recalculando los centroides, para finalmente aplicar la fase correspondiente al criterio de paro del algoritmo.

Los métodos de solución mencionados anteriormente fueron tomados como punto de referencia para el desarrollo del algoritmo propuesto para el CVRP, teniendo claras diferencias con cada uno de ellos, mismas que se mencionan de manera detallada en el punto 4.

Uno de los problemas de optimización más estudiados dentro de las Ciencias Computacionales es el problema de Ruteo Vehicular, por sus siglas en inglés (*Vehicle Routing Problem*), el cual se encuentra clasificado como NP-Completo [4] y algunas de sus variantes, como es el caso del Problema de Ruteo Vehicular con Restricciones de Capacidad en Vehículos (CVRP). Es por su bien conocida complejidad, que este problema ha sido abordado mediante la aplicación de métodos de agrupamiento, los cuales permiten generar grupos de clientes en los cuales se minimice el costo del recorrido al interior cada grupo generado [3], para este problema los métodos más utilizados debido a su eficacia y eficiencia para ciertos tamaños de instancias son el k-means y sus variantes [9, 11].

Los resultados obtenidos a cada problema van a depender del tipo de algoritmo de agrupamiento aplicado, así como de las características propias del problema, ya que cada algoritmo o método de agrupamiento cuenta con ciertas ventajas y desventajas, las cuales deben ser analizadas antes de aplicarlos a un problema en específico, tomando en cuenta el tipo y tamaño de instancias a resolver, debido a que algunos de los métodos de agrupamiento son sensibles a los cambios en la distribución y a la cantidad de datos analizados.

En el presente trabajo se presenta un nuevo enfoque para un método no supervisado, el cual se encarga de generar agrupamientos para el Problema de Ruteo Vehicular con Restricciones de Capacidad en Vehículos, para ello, se lleva a cabo un estudio sobre cada uno de los métodos de agrupamiento que han sido más

comúnmente aplicados a los problemas de Ruteo Vehicular, enfatizando sus características, así como sus ventajas y desventajas, de modo que se incorporan las ventajas de algunos algoritmos al método propuesto, además de minimizar algunas de sus desventajas e incorporar restricciones propias del problema a tratar, con lo cual se obtiene un algoritmo de agrupamiento eficaz.

La estructura del presente trabajo se encuentra de la siguiente manera: En la sección 2 se explica el concepto y complejidad del problema de agrupamiento, además de mencionar las características deseables de los algoritmos de agrupamiento y las características principales de los algoritmos más utilizados para tratar problemas de Ruteo Vehicular; en la sección 3 se presenta una descripción conceptual del problema de Ruteo Vehicular con Restricciones de Capacidad en Vehículos, así como su representación gráfica. En la sección 4 se presenta el algoritmo de agrupamiento propuesto en este trabajo de investigación para el problema tratado; en la sección 5 se presentan las conclusiones referentes al estudio y análisis realizado, para finalmente mostrar los trabajos futuros y las referencias utilizadas en este trabajo de investigación en la sección 6.

2 Problema de Agrupamiento

El problema de generar k agrupamientos de una población heterogénea de N elementos es considerado como un problema NP-Completo [6, 8], cuando este se encuentra bajo ciertas funciones objetivo o características que incrementan su complejidad, aun cuando los puntos a ser agrupados se encuentren en un espacio euclídeo de dos-dimensiones [6]. Para estos casos, los métodos de solución aplicados son los *heurísticos*, debido a que para este tipo de problemas no se conoce un algoritmo determinístico en tiempo polinomial que los resuelva. Una manera formal de definir el problema es la siguiente [8]: Dado $X \in R^{m \times n}$ es un conjunto de elementos representando un conjunto de m puntos x_i en R^n , donde el objetivo es obtener la división de X en K grupos C^k , de modo que cada elemento perteneciente a un mismo grupo cuenta con características similares a los demás elementos de dicho grupo, mostrando diferencias sustanciales con elementos de grupos diferentes. Cada uno de los K grupos obtenidos durante la segmentación se conoce como *grupo* o *agrupamiento*. En la mayoría de los casos, los agrupamientos se realizan en base a la cercanía entre los elementos de una población, tratando de minimizar la distancia entre los elementos de un mismo agrupamiento.

Para tratar el problema de agrupamiento se requiere llevar a cabo un análisis basado en la información característica de los datos evaluados, de modo que permita encontrar las posibles relaciones existentes entre dichos datos (la similaridad se evalúa en base a un criterio o característica establecida en el algoritmo) para lo cual se elige un *centroide* de forma aleatoria, a partir del cual se van identificando las similitudes (distancia o características especificadas en el algoritmo) de los elementos de la población con el *centroide* seleccionado, de modo que teniendo los elementos más similares en un mismo grupo, los cuales cuentan con diferencias considerables con elementos de otros grupos, se obtendrán agrupamientos de mejor calidad. Para ejemplificar el problema de agrupamiento, a continuación (figura 1) se presenta un conjunto de elementos (datos) que se encuentran dispersos en un espacio euclídeo y

para el cual se muestran tres de las posibles soluciones (agrupamientos) a dicha instancia.

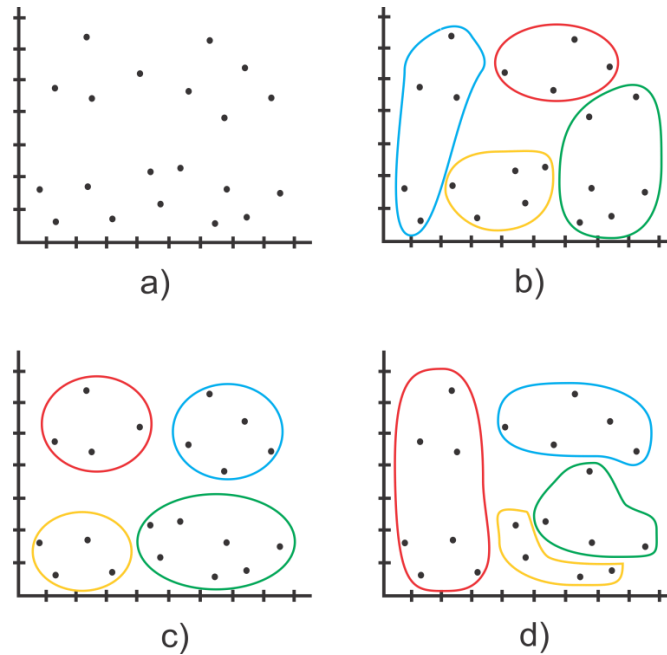


Fig. 1. Algunos de los posibles agrupamientos generados (b, c, d) para una instancia pequeña de 20 elementos (a).

El análisis de agrupamiento llevado a cabo para segmentar un problema, puede ser descrito como una herramienta de descubrimiento, debido a que permite encontrar relaciones que no habían sido consideradas entre los elementos de una población, basándose en la complejidad de los datos [7]. De modo que el número de agrupamientos K a generar en una población, puede ser fijado por el programador o generados de modo que se obtenga el menor número de agrupamientos, lo cual dependerá del análisis de agrupamiento y del tipo algoritmo de agrupamiento implementado.

2.1 Características Deseables en los Algoritmos de Agrupamiento

En la literatura se encuentran especificadas de manera general, las características deseables para los algoritmos de agrupamiento [14].

- **Escalabilidad:** Una gran mayoría de los algoritmos de agrupamiento obtienen buenos resultados para instancias pequeñas, presentando problemas tanto de eficiencia como de eficacia al tratar de resolver instancias grandes (miles o millones de elementos).

- **Habilidad para trabajar con distintos tipos de atributos:** Muchos de los algoritmos existentes fueron desarrollados para trabajar solo con datos numéricos, mientras que existen problemas que requieren el uso de datos binarios, discretos o alfanuméricos.
- **Descubrimiento de grupos con formas arbitrarias.** La mayoría de los algoritmos de agrupamiento se basan en el nodo o elemento más cercano acorde con la distancia euclidiana, con lo que se obtienen grupos circulares y de densidad similar, lo que limita la diversidad de los resultados del algoritmo.
- **Requerimientos mínimos en el conocimiento de los parámetros de entrada.** Gran cantidad de los algoritmos existentes solicitan al usuario conocer el número de grupos a generar, lo cual es una limitante que compromete la calidad del algoritmo, ya que en el caso de instancias de miles o millones de elementos es imposible conocer cuál es el número mínimo de grupos que se pueden generar, además de que la complejidad varía de acuerdo a las características propias del problema tratado.
- **Habilidad para tratar con el ruido.** Algunas poblaciones cuentan con elementos erróneos, de comportamiento extraño o elementos faltantes, a lo cual la mayoría de los algoritmos de agrupamiento son sensibles, lo que compromete la calidad de las soluciones generadas.
- **Insensibilidad al orden de inicio.** Algunos algoritmos son sensibles al orden en que se inicia el análisis de la población. Por ejemplo, para una misma población, dependiendo del elemento con que se inicie va a variar el número de agrupamientos generados, lo cual afecta la calidad en los resultados del algoritmo.
- **Insensibilidad al tipo de distribución.** Una gran mayoría de los algoritmos existentes, funcionan bien para poblaciones que se encuentran agrupadas de forma natural, es decir se encuentran separadas en grupos de acuerdo a su localización, pero su eficacia y eficiencia comienza a disminuir cuando se trabajan con poblaciones de distribución completamente aleatoria, y se incrementa más aún, cuando la distribución de los elementos además de ser agrupada, incorpora elementos dispersos de forma aleatoria.

Los algoritmos tienen comportamientos diferentes dependiendo de las características propias de la población a evaluar, tales como la densidad requerida para los agrupamientos y de la distribución de los datos y parámetros de entrada. Cabe mencionar que las características que debe cumplir un algoritmo se encuentran en función las características del problema a tratar, además de que no todos los algoritmos de agrupamiento son eficaces y eficientes para todos los problemas, por lo que existen diversos algoritmos en la literatura enfocados a problemas de diferentes ámbitos.

3 Problema de Ruteo Vehicular con Restricciones de Capacidad en Vehículos

El Problema de Ruteo Vehicular, por sus siglas en inglés *VRP (Vehicle Routing Problem)*, es un problema combinatorio complejo, clasificado en la literatura como NP-Completo [15, 16].

El VRP con restricciones de Capacidad en Vehículos es un problema de gran importancia, no solo para las ciencias computacionales, sino para diversas áreas, entre ellas la industrial, debido a que aborda uno de los problemas principales del área, como lo es la distribución y logística. El problema radica en definir el menor número de rutas de manera que se minimice el costo total del recorrido, y se puede definir de manera formal como el conjunto de N clientes que cuentan con una demanda d , la cual debe ser satisfecha por uno de K vehículos disponibles, de modo que la suma de la demanda de los clientes asignados a un vehículo k no exceda la capacidad máxima del vehículo C . Cabe mencionar que el punto de partida 0 y regreso $n+1$ de los vehículos, es el depósito. Una vez que todos los clientes han sido asignados y los vehículos involucrados han finalizado sus respectivas rutas, se dice que se ha encontrado una solución al problema. Estas características pueden ser representadas por medio de un modelo de Programación Lineal Entera Binaria propuesto por [17], al cual le relajadas las restricciones de tiempo (Figura 2).

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

Sujeto a:

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

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

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

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

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq C \quad \forall k \in K \quad (6)$$

$$x_{ijk} \geq 0 \quad \forall k \in K, (i,j) \in A \quad (7)$$

$$x_{ijk} \in \{0,1\} \quad \forall k \in K, (i,j) \in A \quad (8)$$

Fig. 2. Modelo de Programación Lineal Entera Binaria para el VRP con restricciones de Capacidad en Vehículos [17]

La instrucción 1 del modelo matemático (Figura 2) representa la función objetivo, la cual es minimizar el costo total del recorrido, y de forma implícita reducir el número de vehículos requeridos para realizar el recorrido total. Las restricciones en 2 especifican que el número de vehículos que atienden a un cliente debe ser igual a uno.

Las restricciones en 3 aseguran que el número de clientes directamente alcanzables por el vehículo k a partir del depósito al inicio del recorrido es igual a uno. El conjunto de restricciones en 4 especifica que el número de vehículos que llegan a un vehículo es el mismo número de vehículos que salen de dicho cliente. Las restricciones en 5 aseguran que para cada vehículo k el número de nodos que conectan con el depósito al final del recorrido es igual a uno. Las restricciones en 6 especifican que la suma de la demanda de todos los clientes asignados a un vehículo k no debe exceder la capacidad máxima del vehículo. La instrucción 7 garantiza la no negatividad de las variables x y finalmente la instrucción 8 define al modelo como un modelo lineal entero binario.

3.1 Representación Gráfica

La representación de una instancia del VRP con restricciones de Capacidad en Vehículos se puede realizar por medio de un grafo disyuntivo formando un cliqué. A continuación, en la Figura 3 se muestra la representación de una instancia de 8 clientes.

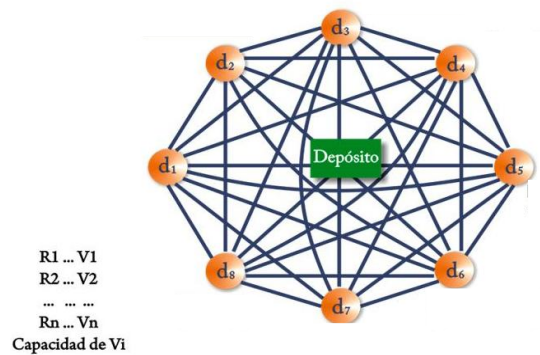


Fig. 3. Grafo disyuntivo para una instancia de 8 clientes.

En el grafo disyuntivo, cada vértice corresponde a un cliente, donde cada uno cuenta con una demanda d la cual debe ser atendida por un vehículo k . Cada vehículo corresponde a una ruta Rn , donde se debe respetar la capacidad máxima del vehículo asignado, teniendo como objetivo general, minimizar el costo total del recorrido, así como el número de vehículos utilizados. A continuación (Figura 4), se presenta una posible solución a la instancia representada por medio de un digrafo.

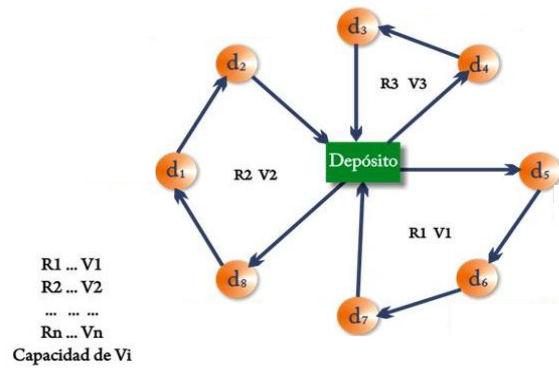


Fig. 4. Representación de una solución mediante un dígrafo para una instancia de 8 clientes.

En el dígrafo correspondiente a una solución (Figura 4), se cumple con las restricciones especificadas en el modelo matemático. En esta solución se definen 3 rutas, mismas que están a cargo de 3 vehículos indicando el orden de atención de los mismos y respetando las restricciones de capacidad de cada vehículo.

3.2 Algoritmos de Agrupamiento Aplicados al CVRP

El VRP con restricciones de Capacidad en Vehículos, debido a su complejidad ha sido abordado por diversos métodos heurísticos, entre los que se encuentran los algoritmos de agrupamiento, los cuales permiten segmentar el problema y abordarlo de manera más sencilla, facilitando la generación de la solución inicial, debido a que la mayoría de los algoritmos de agrupamiento aplicados al problemas de Ruteo Vehicular generan los grupos en base al criterio del cliente más cercano. Otro punto importante es que estos algoritmos también se han implementado para aumentar la exploración del espacio de soluciones.

En este trabajo de investigación se presenta un análisis de los algoritmos de agrupamiento que se aplican más comúnmente a problemas de Ruteo Vehicular, destacando sus ventajas y desventajas.

K-means

Es un algoritmo de particionamiento, que se basa en la minimización de la distancia interna al centroide del grupo. El k-means ha sido el algoritmo de agrupamiento más utilizado en problemas combinatorios debido a su facilidad de implementación y a su relativa eficiencia [18, 19, 20, 21]. Sus desventajas radican en que el algoritmo es totalmente dependiente de la configuración inicial, ya requiere se le especifique el número de grupos a generar, además de que los centroides son generados de forma aleatoria como un punto en el espacio, lo cual implica que tienen que ser recalculados tanto los centroides como las distancias de los clientes, lo cual conlleva una baja en la eficiencia del algoritmo al manejar poblaciones de miles o millones de elementos.

Otro punto a destacar es que el algoritmo muestra debilidad al tratar con grupos de formas arbitrarias o de diferentes tamaños [20], lo cual se ve reflejado en la calidad de las soluciones obtenidas y en la eficiencia del algoritmo.

PAM (Particionamiento a partir de la Mediana)

Variante del k-means que lleva a cabo agrupamientos difusos, particionando una población en cierta cantidad de grupos previamente establecidos. Sus ventajas radican en que es un método más robusto que k-means y de acuerdo a la literatura es muy eficiente para bases de datos pequeñas. Por el contrario, si se requiere el uso de poblaciones muy grandes, este algoritmo no es recomendable, debido a que presenta problemas de escalamiento, lo que compromete la eficiencia del mismo [21, 22, 23].

ISODATA

Es un algoritmo más sofisticado que el k-means, el cual utiliza técnicas de análisis de datos iterativas de auto-agrupamiento. Sus ventajas radican en su flexibilidad debido a que a diferencia del k-means, éste requiere recibir una estimación inicial del número de grupos a generar, los cuales pueden ser ajustados de forma automática, eliminando los grupos pequeños [24], además de generar grupos de densidad similar, por lo que en caso de existir grupos muy grandes, el algoritmo los divide de acuerdo a la desviación estándar existente entre los elementos [24]. Otro punto a destacar es que a diferencia de otros métodos, ISODATA no es sensible a los cambios en la distribución de los elementos. Pero, también cuenta con algunas desventajas, como es que el algoritmo puede volverse muy ineficiente cuando se evalúan instancias muy grandes [24], además de que requiere sintonizar los valores de una serie de parámetros necesarios para llevar a cabo el agrupamiento, como son: número máximo de iteraciones, valor del umbral entre grupos y valor de umbral entre elementos de un mismo grupo, lo cual requiere de un proceso de prueba y error previo a la experimentación [24].

Batchelor y Wilkins

A diferencia del algoritmo ISODATA, el algoritmo propuesto por Batchelor y Wilkins genera agrupamientos en base a un umbral sin necesidad de especificar el número de grupos requerido, mostrando buena eficiencia cuando existen cambios en la distribución de los elementos. Pero tiene una desventaja, ya que el método es muy sensible a los valores del parámetro, por lo que se tiene que realizar un análisis de sensibilidad para determinar el valor adecuado del umbral, de modo que no se comprometa la eficacia de la solución [24].

4 Algoritmo de Agrupamiento Propuesto

En este trabajo de investigación se presenta una propuesta de un Algoritmo de Agrupamiento (CA-CVRP) enfocado a tratar el Problema de Ruteo Vehicular con restricciones de Capacidad en Vehículos, el cual retoma las ventajas de los algoritmos analizados en el punto 3 y minimiza sus desventajas, con la finalidad tener un algoritmo escalable y robusto que genere agrupamientos de buena calidad para el problema tratado.

Las diferencias entre los algoritmos existentes en la literatura y el método propuesto son claras, siendo las más notables con [25], donde se utiliza el algoritmo k-medias en las 3 versiones de algoritmos de agrupamiento propuestos. Cabe mencionar que para el caso de k-medias, este requiere se introduzca el número de agrupamientos a generar, mismo que sucede en el caso de [26] y [27], donde se basan en el algoritmo k-means; desventaja que es mejorada en el algoritmo propuesto.

Otra diferencia consiste en que en el algoritmo propuesto no es necesario recalcular centroides para incorporar clientes no agrupados, contrario a [25, 26 y 27], además de tomar en cuenta restricciones propias del problema para llevar a cabo los agrupamientos, lo cual no se realiza en [26 y 27]; en el caso de [25] se toman en cuenta restricciones de los problemas tratados, para agruparlos, ya que en este caso no existe un costo o cercanía a evaluar, contrario a lo realizado por el algoritmo propuesto, donde toma en cuenta minimizar la distancia al interior de cada grupo, así como la restricción de capacidad del problema, con lo que se obtienen soluciones factibles sin necesidad de aplicar otro método de solución, como sucede en [25, 26 y 27].

4.1 Desarrollo del CA-CVRP

El CA-CVRP (Clustering Algorithm – Capacited Vehicle Routing Problem) es un método heurístico no supervisado, el cual surge del análisis de las ventajas y desventajas de los métodos existentes aplicados al VRP con restricciones de Capacidad en Vehículos, algunos de los cuales muestran deficiencias y debilidades cuando existen cambios en el tipo de distribución, tamaño y forma de los grupos, así como en el tamaño de la población, ya que la mayoría de ellos no son escalables [24, 21, 22, 23]. A continuación se muestra el algoritmo utilizado para el método propuesto (figura 5):

```

1. Inicializar (N, Vh, C)
2. Leer Datos de Entrada
3. para i=0 : i < N
    para j=0 : j < N
         $d[i][j] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 
    fin-para
fin-para
4. Inicializar Estructura Solution
   vect_tabu, index
5. Repetir
   centroide = 1+(rand()%N)
   si vect_tabu[centroide] == 1 entonces
       Repetir centroide = 1+(rand()%N)
       Hasta vect_tabu[centroide] == 0
   fin-si
   si-no
       vect_tabu[centroide] == 1
       Busca_cliente_mas_cercano_al_centroide
       si  $(d_i \cdot \sum_{k=1}^n d_k) \leq C$  entonces
           si vect_tabu[i] == 1 entonces
               Repetir Busca_cliente_mas_cercano_al_centroide
               Hasta vect_tabu[i] == 0
           si  $(d_i \cdot \sum_{k=1}^n d_k) > C$  entonces v++ fin-si
       fin-si
   fin-si
   si-no
       vect_tabu[i] == 1
       Solution[index] = i
   fin si-no
fin si-no
Hasta index == N-1
6. Imprime Agrupamiento

```

Fig. 5. Algoritmo del CA-CVRP

El algoritmo inicialmente lee el benchmark de entrada, para posteriormente llevar a cabo el cálculo de las distancias euclidianas correspondientes a los clientes y al depósito, cuyos valores se calculan con la fórmula 1 [28] y son almacenados en una matriz cuadrada de $N \times N$.

$$d_{i,j} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Posteriormente se selecciona un *centroide* de forma aleatoria. Cabe mencionar, que a diferencia de otros métodos de agrupamiento, para este algoritmo un *centroide* es un cliente, lo que evita el trabajo de recalculer distancias y *centroides* en cada iteración, como es el caso del k-means. Una vez elegido el *centroide*, el algoritmo verifica que el cliente elegido no haya sido asignado a otra ruta, si esta condición se cumple, se

toma el *centroide* y se inicia la búsqueda del cliente más cercano al *centroide* elegido y que no haya sido previamente asignado.

Un punto importante a destacar, es que el algoritmo no solo se enfoca en agrupar los clientes más cercanos a un *centroide* dado, sino que también evalúa la restricción correspondiente a la demanda de los clientes y capacidad máxima del vehículo, de modo que si el cliente seleccionado excede la capacidad máxima del vehículo actual, dicho cliente es descartado y se continúa en la búsqueda de otro cliente que pueda ser asignado al grupo, de lo contrario, se genera otra ruta, hasta que todos los clientes hayan sido asignados a un vehículo.

El enfoque manejado por el algoritmo propuesto, evita recalcular las distancias a cada paso, de modo que se reduce el tiempo computacional, además de que no es necesario conocer el número de grupos a generar, debido a que el algoritmo minimiza el total de rutas utilizadas, además de tomar en cuenta la distancia dentro de cada grupo.

5 Pruebas Experimentales

Para probar la eficacia del CA-CVRP se llevaron a cabo pruebas experimentales utilizando benchmarks de 100 y 1000 clientes respectivamente, en sus tres diferentes tipos de distribución, las cuales son: Agrupados, Aleatorios y Agrupados-Aleatorios, esto con la finalidad de probar la escalabilidad e insensibilidad del algoritmo a los cambios en el tipo de distribución.

Las pruebas fueron llevadas a cabo en una computadora portátil con un procesador Quad core Intel i7 de 1.73 – 2.93 GHz, 6 GB en RAM y desarrollado sobre Visual C 2008.

De los benchmarks mencionados anteriormente, se utilizaron cuatro instancias, tres de 100 clientes, cada una correspondiente a un tipo de distribución (C101, R101 y RC101) y una de 1000 clientes, perteneciente a la distribución, que de acuerdo a la literatura es la más difícil de tratar, que es la RC (RC110_1). Cada una de estas instancias fue ejecutada 30 veces. Los resultados obtenidos corresponden a los agrupamientos generados para cada instancia, los cuales fueron evaluados exhaustivamente en cuanto a la factibilidad. Un ejemplo de los resultados obtenidos para cada una de las instancias se muestra a continuación en las figuras 6a (a, b, c) y 6b.

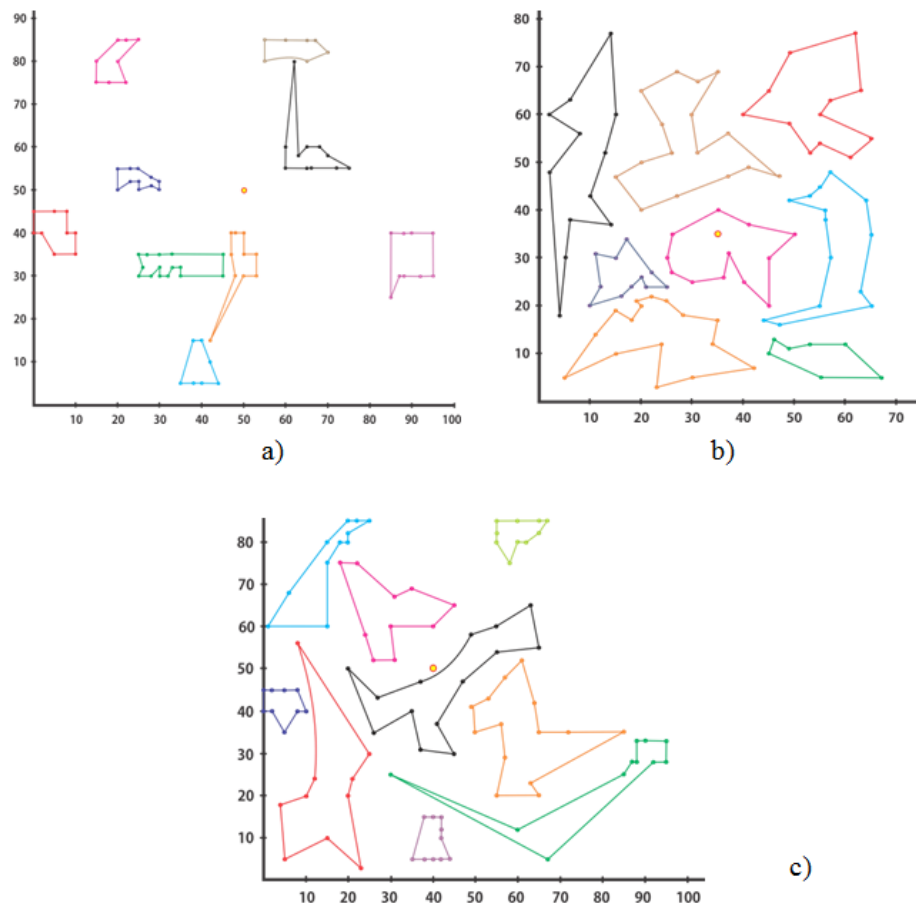


Fig. 6a. Ejemplo de uno de los resultados obtenidos con el CA-CVRP para la instancia a) C101, b) R101, c) RC101

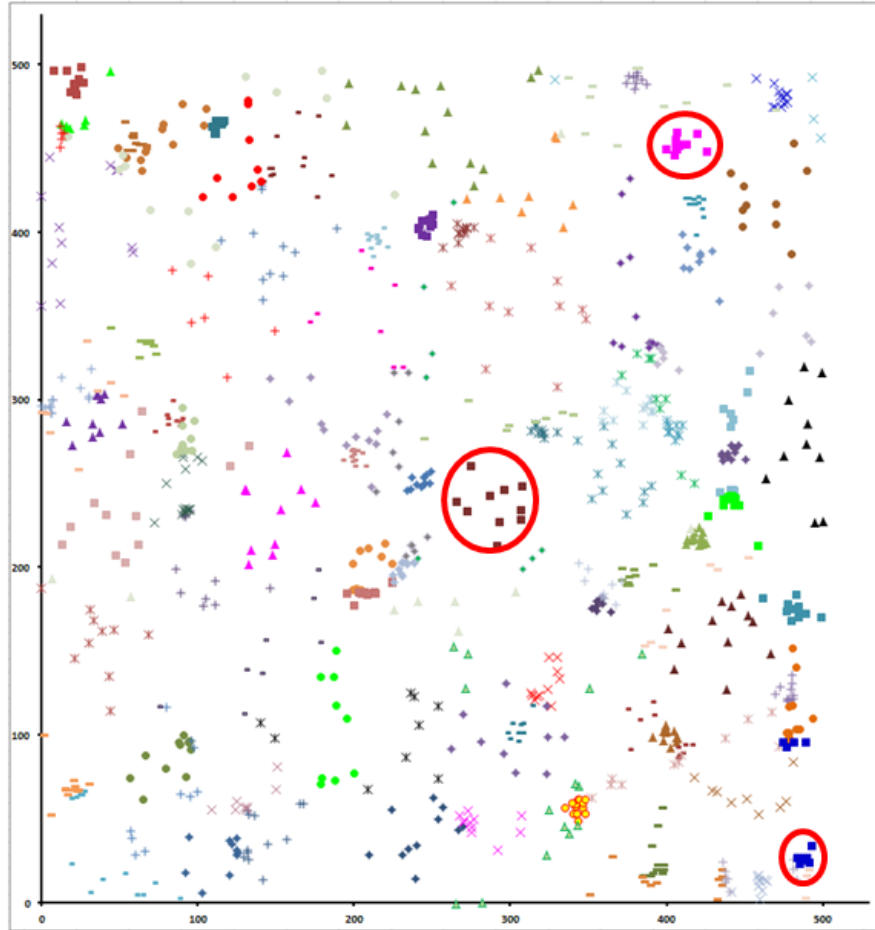


Fig. 6b. Ejemplo de uno de los resultados obtenidos con el CA-CVRP para la instancia RC110_1

De acuerdo a los resultados obtenidos para las instancias de 100 y 1000 clientes, se puede observar que se están generando agrupamiento de buena calidad, debido a que no se tienen clientes aislados (outliers) debido a las restricciones que se están evaluando en el algoritmo (no solo los casos en donde los clientes se encuentran alejados de acuerdo a su localización), además de que el tiempo para generar una solución es mínimo, por lo que habla de un algoritmo rápido. A continuación (tabla 1), se muestra los resultados promedio obtenidos de las 30 ejecuciones de cada instancia. Se realizaron 30 ejecuciones por cada instancia, debido a que para este tipo de experimentación es común utilizar 30 experimentos como tamaño de muestra mínimo aceptable [29].

Tabla 1. Promedio de los resultados obtenidos para cada instancia ejecutada

<i>Benchmark</i>	<i>Num_Agrupamientos</i>	<i>Tiempo (segs.)</i>
C101	9	$6 * 10^{-5}$
R101	8	$6 * 10^{-5}$
RC101	9	$7 * 10^{-5}$
RC110_1	94	$6 * 10^{-3}$

De acuerdo a los resultados obtenidos, se observa que el algoritmo cumple con el objetivo planteado, ya que muestra un muy buen desempeño para cualquier tamaño de instancia, siendo insensible a los cambios en el tipo de distribución, además de mostrar mejoras considerables a las desventajas presentadas por algoritmos existentes. Las ventajas presentadas por el algoritmo al realizar pruebas preliminares fueron:

- No es necesario conocer el número de grupos a generar
- Insensible a los cambios en el tipo de distribución
- Escalabilidad
- Minimiza la distancia entre los elementos de cada grupo
- No recalcula distancias
- Incorpora la restricción de capacidad del VRPTW
- Eficaz y Rápido

Se menciona que el CA-CVRP es rápido, debido a que en las pruebas experimentales con 100 clientes se obtuvo una solución en $7 * 10^{-5}$ seg., mientras que para instancias de 1000 clientes tardó $6 * 10^{-3}$ seg., obteniendo agrupamientos de buena calidad. Cabe destacar que los benchmarks utilizados fueron los de Solomon y Gehring y Homberger, utilizando los tres tipos de distribución manejados, para los cuales se obtuvieron resultados similares en calidad y rapidez. En este caso no se está hablando de la eficiencia, debido a que es necesario realizar pruebas comparativas con otros algoritmos de agrupamiento para comprobar la eficiencia del algoritmo propuesto.

Conclusiones

En este artículo se presenta el algoritmo CA-CVRP, el cual fue desarrollado específicamente para tratar el problema de agrupamiento del problema de Ruteo Vehicular con restricciones de Capacidad en Vehículos, basado en el enfoque del cliente más cercano y evaluando la restricción de capacidad propia del problema.

CA-CVRP utiliza un enfoque que evita recalcular distancias en cada iteración, donde se consideran los *centroides* como clientes y no como puntos en el espacio, otra ventaja sobresaliente es que el algoritmo no requiere que se le especifique el número de agrupamientos a realizar, a diferencia de los trabajos presentados en [25, 26 y 27], además de que no necesita de otra heurística para evaluar las restricciones del problema, debido a que los agrupamientos se llevan a cabo tomando en cuenta la característica de capacidad en vehículos.

De acuerdo a pruebas experimentales realizadas con benchmarks de 100 y 1000 clientes utilizando tres diferentes tipos de distribución, se obtuvieron resultados eficaces en un tiempo mínimo, además de que el algoritmo demostró ser robusto, ya que no se necesita conocer el número de agrupamientos a generar, no muestra sensibilidad a diferentes tipos de distribución, además de ser escalable y reducir el tiempo de cómputo al evitar recalcular distancias.

Trabajos Futuros

Realizar el análisis comparativo del CA-CVRP con otros algoritmos de agrupamiento, como el k-means, PAM, ISODATA y el algoritmo de Batchelor y Wilkins, los cuales han sido aplicados al VRP, para lo cual se utilizarán los benchmarks de Solomon de 100 clientes y Gehring y Homberger de 1000 clientes, tomando en cuenta los tres tipos de distribución (C, R y RC), para probar la eficiencia del algoritmo propuesto.

Referencias

- [1] Ganesh K., Dhanalakshmi R., Tangavelu A., Parthiban P. Hybrid Artificial Intelligence Heuristics and Clustering Algorithm for Combinatorial Asymmetric Traveling Salesman Problem. Utilizing Information Technology Systems across Disciplines: Advancements in the Application of Computer Science. pp. 1 – 36. ISBN13: 9781605666167, ISBN10: 1605666165, EISBN13: 9781605666174. 2009.
- [2] Bozdag Doruk. Graph Coloring and Clustering Algorithms for Science and Engineering Applications. Tesis para obtener el Grado de Doctorado en la Universidad del Estado de Ohio. USA, 2008.
- [3] Nallusamy R., Duraiswamy K., Dhanalakshmi R. Pathiban P. Optimization of Multiple Vehicle Routing Problems Using Approximation Algorithms. International Journal of Engineering Science and Technology. Vol. 1(3). pp. 129 – 135. ISSN. 0975-5462. 2009.
- [4] Lenstra J. K., Rinnooy Kan A. H. G. Complexity of Vehicle Routing and Scheduling Problems. Networks. Vol. 11, Issue: 2. ISSN. 00283045. pp. 221 -227. 1981.
- [5] Tan, Steinbach, Kumar. Data Mining Cluster Analysis: Basic Concepts and Algorithms. Lecture Notes for Chapter 8. Introduction to Data Mining. pp. 487-488. Stanford University, 2009.
- [6] González Teofilo F. On the Computational Complexity of Clustering and Related Problems. System Modeling and Optimization. Lecture Notes in Control and Information Sciences. Volume 38/1982, 174-182. 1982.
- [7] Anderberg M. R. Cluster Analysis for Applications. Academic Press, New York, 1973.
- [8] Fung Glenn. A Comprehensive Overview of Basic Clustering Algorithms. IEEE Citeseer. pp. 1 - 37. 2001.
- [9] Nallusamy R., Duraiswamy K., Dhanalakshmi R. and Parthiban P. Optimization of Multiple Vehicle Routing Problems Using Approximation Algorithms. International Journal of Engineering Science and Technology. Vol 1 (3). pp. 129 -135. ISSN. 0975-5462. 2009.

- [10] Kanungo Tapas, Netanyahu Nathan S., Wu Angela Y. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis on Machine Intelligence. Vol. 24, No. 7. pp. 881 – 892. 2002.
- [11] Wu Chih Sheng. A Study of Heuristic Algorithms for Optimization and Clustering Problems. Tesis para Obtener el Grado de Maestría en el Departamento de Manejo de la Información. Universidad Tecnológica de Chaoyang. Taichung, Taiwan, República de China, 2003.
- [12] Mohamed Jafar O. A., Sivakumar R. Ant-based Clustering Algorithms: A Brief Survey. International Journal of Computer Theory and Engineering, Vol. 2, No. 5. pp. 1793 – 8201. 2010.
- [13] Jain A. K., Murty M. N., Flynn P. J. Data Clustering: A Review. ACM Computing Surveys. Vol. 31. No. 3. pp. 264 – 323. 1999.
- [14] Hernández Valadez Edna. Algoritmo de Clustering Basado en Entropía para Descubrir Grupos en Atributos de Tipo Mixto. Tesis para obtener el grado de Maestro en Ciencias con opción en Computación. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. Departamento de Ingeniería Eléctrica Sección Computación. 2006.
- [15] Ochi Luiz S., Vianna Dalessandro S., Drummond Lucia M. A., Victor André O. A Parallel Evolutionary Algorithm for the Vehicle Routing Problem with Heterogeneous Fleet. Parallel and Distributed Processing. Lecture Notes in Computer Science. Vol. 1388 / 1998. pp. 216 – 224. Computer Science, 1998.
- [16] Chin. A., Kit H., Lim A. A New GA Approach for the Vehicle Routing Problem. Proceedings of 11th. IEEE International Conference on Tools with Artificial Intelligence. ISSN: 1082-3409. ISBN. 0-7695-0456-6. pp. 307 – 310. 2002.
- [17] Toth Paolo, Vigo Daniele. The Vehicle Routing Problem. ISBN. 0-89871-579-2. pp. 158 – 159. Ed. Siam, 2002.
- [18] Fu Jie, Lei Lin, Zhou Guohua. A Parallel Ant Colony Optimization Algorithm with GPU-Aceleration Based on All-in-Roulette Selection. Third International Workshop on Advanced Computational Intelligence. China, 2010.
- [19] Mount David M. Kmlocal: A Testbed for k-means Clustering Algorithms. University of Maryland and David Mount. Partially Supported by the National Science Foundation. 2005.
- [20] Karypis George, Hong Eui, Kumar Vipin. CHAMALEON: A Hierarchical Clustering Algorithm using Dynamic Modeling. IEEE Computer. Vol. 32. pp. 68 – 75. ISBN. 0018-9162. 1999.
- [21] Stan Salvador, Chan Philip. Determining the Number of Clusters / Segments in Hierarchical Clustering / Segmentation Algorithms. 16th. IEEE International Conference on Tools with Artificial Intelligence. ISBN. 0-7695-2236-X. 2004.
- [22] Huang Zhe Xue. Extensions to the k-means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery. pp. 283 – 304. Netherlands, 1998.
- [23] Wu Chih-Sheng. A Study of Heuristic Algorithms for Optimization and Clustering Problems. Tesis para obtener el grado de Maestría en Administración de la Información. China, 2003.
- [24] Memarsadegui Nargess, Mount David M., Netanyahu Nathan S., Moigne Jacqueline Le. A Fast Implementation of the Isodata Clustering Algorithm. International Journal of Computational Geometry & applications. pp. 71 – 103, 2007.
- [25] Cruz Reyes Laura. Clasificación de Algoritmos Heurísticos para la Solución de Problemas de Bin Packing. Tesis para obtener el grado de Doctor en Ciencias de la Computación. Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet). Junio 2004.

- [26] Lian L. Castelain E. A Decomposition-based Heuristic Approach to Solve General Delivery Problem. Proceedings of the World Congress on Engineering and Computer Science. Vol. II. ISBN:978-988-18210-2-7. WCECS 2009, San Francisco, USA. 2009.
- [27] Díaz Parra Ocotlán, Ruiz Vanoye Jorge A., Zavala Díaz José C. Population Pre-selection Operators used for Generating a Non-random Initial Population to Solve Vehicle Routing Problem with Time Windows. Scientific Research and Essays Vol. 5 (22). pp. 3529 – 3537. ISBN. 1992-2248. Academic Journal 2010.
- [28] Krajewski Lee J., Ritzman Larry P. Administración de Operaciones: Estrategia y Análisis. 5ta Edición. Vol. 1. Ed. Addison-Wesley Iberoamericana. pp. 376. ISBN. 9789684444119. México, 2000.
- [29] Sheldon M. Ross. Simulación. 2da. Ed. Ed. Prentice Hall. 1999.