

# Un Mecanismo de Vecindad con Búsqueda Local y Algoritmo Genético para el Problema de Transporte con Ventanas de Tiempo

Marco Antonio Cruz-Chávez<sup>1</sup>, Ocotlán Díaz-Parra<sup>1</sup>, Crispín Zavala-Díaz<sup>2</sup>, Eric Barreto-Sedeño<sup>1</sup>, Martín G. Martínez-Rangel<sup>1,2</sup>, María Guadalupe Quevedo-Alvarez<sup>1</sup>.

<sup>1</sup>CIICAp, <sup>2</sup>FCAeI, Universidad Autónoma del estado de Morelos, Avenida Universidad 1001. Col. Chamilpa, C.P. 62210. Cuernavaca, Morelos, México.  
{mcruz, odiazp, crispin\_zavala, mmtzr, mgquevedo}@uaem.mx  
Ericsson\_98@hotmail.com

**Resumen.** Las técnicas de búsqueda por vecindad han resultado medios útiles para encontrar soluciones aproximadas a problemas de optimización combinatoria. Una vecindad está definida como el conjunto de soluciones cercanas a una solución inicial dada. En este artículo se presenta el diseño de un mecanismo de vecindad combinado con un algoritmo genético. Este diseño híbrido se propone con la finalidad de explotar el espacio de soluciones del problema del transporte con ventanas de tiempo. La vecindad se propone con movimientos tipo uno-óptimos. En este artículo se muestra la etapa de análisis y diseño de la estructura de vecindad con búsqueda local combinada con un algoritmo genético para el problema de transporte con ventanas de tiempo.

## 1 Introducción

Las técnicas de búsqueda por vecindad comúnmente se llaman algoritmos de búsqueda local. La parte medular en el éxito de una búsqueda en vecindad es su estructura y tamaño. En cuanto al tamaño de la vecindad mientras más grande sea mayor será la calidad de las soluciones. Una vecindad amplia produce una heurística más eficaz. La tendencia de utilizar procedimientos de búsqueda local en combinación con otras estrategias se refleja en los trabajos encontrados en literatura por ejemplo: Búsqueda Local Guiada de Vondouris y Tsang [1], GRASP de Feo y Resende [2] [3] y Concentración Heurística (Fase I) de Rosing y Reville [4]. Estas estrategias son en realidad repeticiones de procedimientos de Búsqueda Local (donde se modifica la función objetivo y la solución inicial se reemplaza por una mejor); los Algoritmos Meméticos según Moscato [5], añaden a las operaciones de los Algoritmos Genéticos la Búsqueda Local; también aparece en algunas fases de intensificación en Búsqueda Tabú como menciona Glover [6][7] y en Búsqueda Dispersa el trabajo de Laguna [8]; la Búsqueda Tabú básica o el recocido Simulado por Kirkpatrick [9][10] son en realidad cadenas de movimientos vecinales.

La tendencia es aplicar búsqueda local en vecindades junto con otras técnicas de solución a problemas de optimización combinatoria [17]

En este artículo se propone un mecanismo de vecindad inmerso en un algoritmo genético con la finalidad de realizar una explotación del espacio de soluciones del problema de transporte con ventanas de tiempo. El problema del transporte o VRP (Vehicle Routing Problem) como se conoce en literatura [12], es un problema que se le da principal importancia en las áreas de transportación, distribución y logística. En algunos sectores de la industria, la transportación significa un alto porcentaje de valor agregado a los productos. Por eso la utilización de métodos computacionales para que la transportación ofrezca buenos resultados es de gran utilidad, los ahorros van desde un 5% a un 20% en el total de costos, como lo reporta Toth & Vigo [12]. El problema de transporte con ventanas de tiempo VRPTW (Vehicle Routing Problem with Time Windows) es una variante del VRP y consiste básicamente en minimizar los costos de transportación sujeto a restricciones de tiempo de cada ruta y de capacidad en base a la demanda de cada cliente [12]. La función objetivo que representa a este problema se describe en la ecuación (1) y las restricciones de ventana de tiempo en la ecuación (2).

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

$$a_i \sum_{j \in \nabla_+(i)} x_{ijk} \leq w_{ik} \leq b_i \sum_{j \in \nabla_+(i)} x_{ijk}, \forall k \in K, i \in N \quad (2)$$

Dónde:  $k$ = vehículo,  $i$ =nodo origen,  $j$ =nodo destino,  $x$ =ruta,  $w_i$ =ventana de tiempo del nodo  $i$ ,  $b_i$ = extremo final de la ventana perteneciente al nodo  $i$ ,  $a_i$ =extremo inicial de la ventana perteneciente al nodo  $i$ ,  $\nabla_+(i)$  =conjunto de vértices  $j$  tales que  $(i,j) \in A$ .

Una de las alternativas de solución a este tipo de problemas son los llamados métodos heurísticos. Un método heurístico es un procedimiento para resolver un problema matemático bien definido mediante aproximaciones en un tiempo de computación razonable sin garantizar la optimalidad [13]. El método heurístico utilizado como alternativa de solución para este trabajo es un algoritmo genético aplicado al problema de transporte con ventanas de tiempo. Un algoritmo genético (GA) es un método adaptativo que puede usarse para resolver problemas de búsqueda y optimización. Estos algoritmos están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. Haciendo una analogía del algoritmo genético con el problema del transporte un individuo formado por cromosomas y genes corresponde a una ruta formada por vehículos y nodos. Una vez puesto el marco de referencia de los elementos a utilizar en este artículo, en la sección 2 se explicara a detalle la definición de vecindad, en la sección 3 se explica la vecindad propuesta, en la sección 4 se muestra un análisis del nivel de aplicación de la vecindad en el algoritmo genético, en la sección 5 los trabajos futuros derivados de el análisis presentado en este documento y por último en la sección 6 las conclusiones.

## 2 Definición de Vecindad

Formalmente la vecindad de una solución se define como el conjunto de todas aquellas soluciones que pueden ser alcanzables a partir de una solución  $s'$  por medio de un movimiento  $\sigma$  [11], un movimiento puede ser un intercambio entre elementos que conforman la solución  $s$ .

$$N(s) = \{s' \in S : s \xrightarrow{\sigma} s'\} \quad (3)$$

Donde  $N(s)$  representa la vecindad con respecto a  $s$ ,  $s$  representa una solución tomada del espacio total de soluciones  $S$  y  $s'$  representa el vecino de  $s$  generado a partir de  $\sigma$  movimientos. Un movimiento puede ser una inserción, eliminación o intercambio de componentes en una solución. Para este caso en particular un movimiento estará definido como un intercambio de dos genes en un individuo. En la figura 1 se muestra la representación gráfica de la vecindad generada a partir de una solución  $s$ .

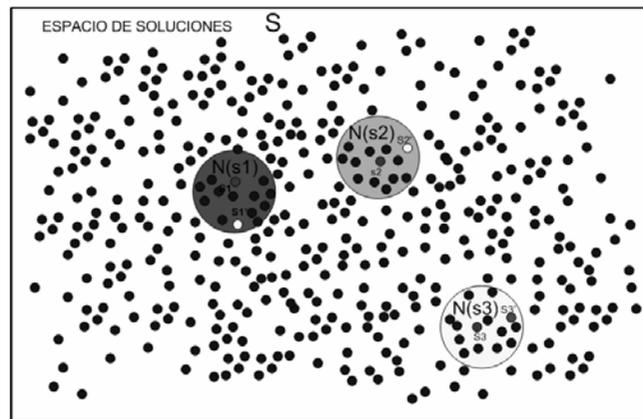


Fig. 1. Vecindad generada a partir de individuo solución  $s$

Por ejemplo la vecindad generada por la solución  $s1$  está representada en la figura 1 como  $N(s1)$  en el círculo gris oscuro y sus vecinos están representados por  $s1'$  y así sucesivamente para cada una de las soluciones  $s_i$  que pertenecen al espacio de soluciones  $S$ , en notación de conjuntos se expresa  $S = \{s_1, s_2, s_3, \dots, s_i\}$ . La estructura de vecindad mediante una búsqueda permite ir haciendo una explotación en el espacio de soluciones. A continuación se explica a detalle cómo se construye la vecindad, los criterios aplicados en la construcción y su justificación.

## 3 Construcción de la Vecindad propuesta

La vecindad que se propone implementar al algoritmo genético para el problema del transporte se basa en la idea propuesta por Or [14], para el Problema del Viajante o

TSP. El método de intercambio de Or es una variante de los conocidos intercambios r-óptimos desarrollados por Lin [15] y Lin & Kernighan [16] para el TSP simétrico. Un intercambio r-óptimo consiste básicamente en mover cadenas de r elementos. Por ejemplo sea 0-1-2-3-4-0-5-6-7-0-8-9-10-0 una ruta válida para un problema de rutas. Un intercambio 2-óptimo consiste en intercambiar cadenas de dos números; entonces para la cadena inicial dada se intercambiarán las cadenas 3-4 y la cadena 8-9 de tal forma que la siguiente ruta factible estará definida por: 0-1-2-**8-9**-0-5-6-7-0-**3-4**-10-0. Para este trabajo se propone utilizar la técnica de Or con 1-óptimo tomando solo movimientos hacia adelante como lo propone Pacheco y Delgado [17] y en este caso para VRPTW se establecerá un límite en el tamaño de la cadena a intercambiar; el tamaño será de uno. En VRPTW cada movimiento en la ruta total implica n-búsquedas para no violar las restricciones propias del problema, por lo que es conveniente utilizar movimientos con cadenas de uno, a esta nueva adecuación la llamamos vecindad tipo OR modificada.

La vecindad propuesta en este análisis es la vecindad tipo OR modificada tomando dos nodos con intercambio de cadenas 1-óptimo y el movimiento se realiza hacia adelante. Y la búsqueda en la vecindad es una búsqueda local clásica. El tamaño de la vecindad se establece mediante la razón del tamaño del individuo y el número de genes a intercambiar, dicha razón se conoce en optimización combinatoria como el número de combinaciones sin repetición de  $m$  elementos con tamaño  $n$  como se representa en la ecuación cuatro.

$$c_m^n = \binom{m}{n} = \frac{m!}{n!(m-n)!} \quad (4)$$

$$T_{N(s)} = c_m^n \quad (5)$$

Dado este análisis se define a  $T_{N(s)}$  como el tamaño de la vecindad generada a partir de  $s$ . Como se muestra en la ecuación 5.

Normalizando las expresiones en la ecuación 6, se tiene que el tamaño de la vecindad  $T_{N(s)}$  con respecto a la solución  $s$  está definido como la razón del tamaño del individuo y del número de genes a mutar esto es:

$$T_{N(s)} = \frac{t_{(s)}!}{n_{(s)}!(t_{(s)} - n_{(s)})!} \quad (6)$$

Donde  $t_{(s)}$  representa el tamaño del individuo y  $n_{(s)}$  representa el número de genes a mutar. Por ejemplo para un individuo formado por 10 genes con intercambio de dos genes,  $t_{(s)} = 10$  y  $n_{(s)} = 2$  por lo tanto  $T_{N(s)} = 45$  lo que implica que solo 45 vecinos serán generados.

El criterio de paro de la vecindad se propone de inicio como lo proponen en literatura dos veces el tamaño de la vecindad como se representa en la ecuación 7.

$$P(N(s)) = 2T_{N(s)} \quad (7)$$

Donde  $P_{(N(S))}$  representa el criterio de paro de la búsqueda en la vecindad  $N(s)$ . Cuando se lleve a cabo la implementación y la experimentación se analizará si es necesario aumentar o disminuir el criterio de paro, de inicio se propone dos veces el tamaño. Una vez identificado el tipo de vecindad, el tamaño de la vecindad y el criterio de paro el siguiente paso es analizar en que parte del algoritmo genético se aplicará la vecindad.

#### 4 Nivel de aplicación de la vecindad sobre el algoritmo genético con búsqueda local

El análisis para la aplicación de la vecindad implica dos partes. La primera consiste en analizar una solución mediante un algoritmo genético para el problema de transporte con ventanas de tiempo y la segunda, mediante este análisis proponer el nivel de aplicación de la vecindad. Por la naturaleza del algoritmo genético se hace uso de una población inicial, sometida a una serie de operadores genéticos como es el operador de selección, el operador de cruzamiento y el operador de mutación como lo muestra la figura 2.

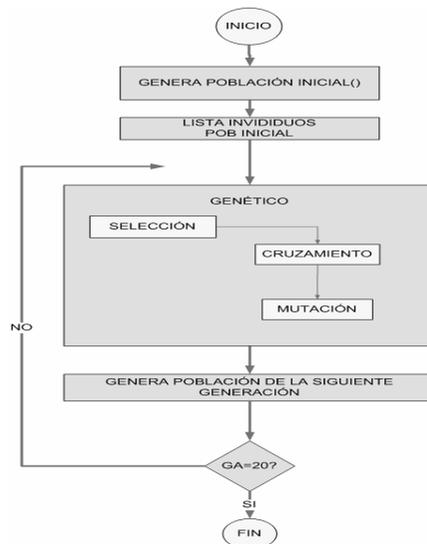


Fig. 2. Diagrama de flujo del algoritmo genético

El algoritmo genético se ejecutó una serie de 10 veces obteniendo los tiempos totales del algoritmo mostrados en la tabla1 y una serie de 10 veces tomando los tiempos por función mostrados en la tabla2.

**Tabla 1.** Tiempos de ejecución del algoritmo genético en serie de 10 para instancias C101 para 100 nodos .

Instancia	Tiempo(seg)	Instancia	Tiempo(seg)
C101-100	2223	C101-100	2131
C101-100	2734	C101-100	2048
C101-100	2196	C101-100	2051
C101-100	2515	C101-100	1985
C101-100	2313	C101-100	2133

En la tabla 1, se muestran los tiempos totales de ejecución, en base a estos datos se obtiene el tiempo de ejecución del algoritmo genético en promedio 2232 segundos. La hipótesis que se plantea es minimizar mediante la estructura de vecindad el tiempo total para hacer más eficiente el algoritmo genético.

En la tabla 2 se muestra la experimentación realizada por función lo que se pretende es detectar cual de las funciones consume mayor tiempo y tomar ese tiempo como parámetro para analizar en específico esa función y analizar si es posible implementar la estructura de vecindad.

**Tabla 2.** Tiempos de ejecución del algoritmo genético por función en serie de 10 para instancias C101 para 100 nodos.

Instancia	Función	Tiempo(seg)	Instancia	Función	Tiempo(seg)
C101-100	Población()	329	C101-100	Población()	345
	Selección()	455		Selección()	490
	Cruzamiento()	543		Cruzamiento()	528
	Mutación()	598		Mutación()	635
	Nva-Generac()	212		Nva-Generac()	215
C101-100	Población()	351	C101-100	Población()	367
	Selección()	523		Selección()	509
	Cruzamiento()	578		Cruzamiento()	554
	Mutación()	667		Mutación()	675
	Nva-Generac()	212		Nva-Generac()	217
C101-100	Población()	357	C101-100	Población()	390
	Selección()	567		Selección()	522
	Cruzamiento()	563		Cruzamiento()	578
	Mutación()	689		Mutación()	656
	Nva-Generac()	213		Nva-Generac()	211
C101-100	Población()	333	C101-100	Población()	346
	Selección()	556		Selección()	512

**Continuación Tabla 2.** Tiempos de ejecución del algoritmo genético por función en serie de 10 para instancias C101 para 100 nodos.

Instancia	Función	Tiem- po(seg)	Instancia	Función	Tiem- po(seg)
C101-100	Cruzamiento()	575	C101-100	Cruzamiento()	524
	Mutación()	636		Mutación()	669
	Nva- Generac()	216		Nva- Generac()	218
	Población()	369		Población()	378
	Selección()	515		Selección()	509
	Cruzamiento()	529		Cruzamiento()	548
	Mutación()	662		Mutación()	649
	Nva- Generac()	216		Nva- Generac()	215

En la tabla 2 se presentan los resultados de tiempos de ejecución en serie de 10 por función, notando que en la función de mutación es la que implica mayor tiempo. Esto nos da la pauta para proponer la estructura de vecindad en lugar del operador de mutación del algoritmo genético como lo muestra la figura 3. Cabe mencionar que en el algoritmo genético la mutación se aplica a nivel de individuo, lo que significa que los movimientos realizados dentro de este operador en el algoritmo genético implica lo que en vecindades se conoce como movimiento vecinal. Entonces lo que se busca es optimizar el tiempo en realizar ese tipo de movimientos. Mediante la propuesta realizada por los autores de este artículo.

EL algoritmo de generación y búsqueda en vecindad se muestra en la figura 4. El algoritmo de generación y búsqueda en vecindad se probó para un individuo de tamaño 10 con un tamaño de vecindad de 45 vecinos generados con el fin de probar la funcionalidad del algoritmo. Una vez probada la funcionalidad se implementará para instancias grandes y por último como trabajo futuro, implementarlo en el algoritmo genético.

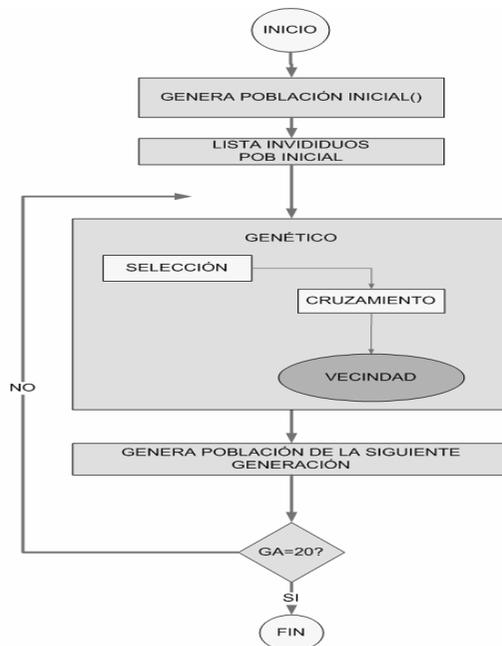


Fig. 3. Nivel de implementación de la estructura de vecindad en el algoritmo genético

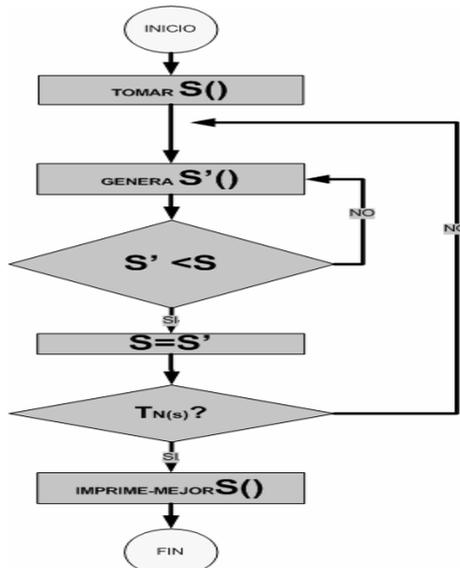


Fig. 4. Algoritmo de generación y búsqueda en vecindad tipo Or-modificada

Como lo muestra la figura4 el algoritmo de vecindad parte de una solución  $s$  tomada del espacio de soluciones y a partir de esa solución se genera una vecindad de tamaño

$T_{N(s)}$  al generar un nuevo vecino  $s'$  se va comparando si es mejor al actual si lo es entonces lo reemplaza como el mejor vecino y así sucesivamente hasta terminar con todos los vecinos generados a partir de la solución. En la figura 5 se muestra el mecanismo de cómo se genera un vecino.



**Fig. 5.** Generación de un vecino

Como ejemplo para ilustrar se toma un individuo formado por 10 nodos y se inicia con el individuo 0 se generan dos aleatorios y se realiza el cambio, para el siguiente vecino se vuelve a tomar el individuo 0 y se vuelven a generar diferentes par de aleatorios y se realiza el cambio así sucesivamente hasta la condición de paro. En el proceso se va comparando si el vecino generado es mejor que el anterior se realiza el cambio; la comparación se realiza en base a la función de evaluación. En este caso la función objetivo a evaluar es la del problema del transporte con ventanas de tiempo. Para el propósito de este documento se propone solo el análisis y el diseño de la estructura.

## 5 Trabajos futuros

Con miras a probar la propuesta expresada en este documento se pretende realizar la implementación del algoritmo vecindario al algoritmo genético y realizar la experimentación y comparación para detectar que tan buena o no resultó la estructura de vecindad aplicada a un algoritmo genético para el problema del transporte con ventanas de tiempo. Se construirá el análisis de tiempos de ejecución del algoritmo genético contra el algoritmo genético con vecindad.

## 6 Conclusiones

El análisis realizado en este documento sirve de punto de partida para conseguir minimizar el tiempo de ejecución de un algoritmo heurístico aplicado a un problema de optimización combinatoria mediante una vecindad. El análisis de tiempos realizado al algoritmo genético para el problema del transporte con ventanas de tiempo generó

resultados que ayudaron a identificar el nivel en que el algoritmo consume mayor tiempo de ejecución. Dicho análisis sirvió para identificar donde se aplicará la estructura de vecindad.

## Referencias

1. VONDOURIS, C. and TSANG, E.,: "Guided Local Search for the Traveling Salesman Problem". *European Journal of Operations Research*. Vol. 113, pp 469-499. (1999).
2. FEO, T. A. and RESENDE, M. G. C.: "A Probabilistic heuristic for a computationally difficult Set Covering Problem". *Operations Research Letters*. Vol, 8, pp 67-71. (1989).
3. FEO, T. A. and RESENDE, M. G. C.: "Greedy Randomized Adaptive Search Procedures". *Journal of Global Optimization*. Vol. 2, pp 1-27. (1995).
4. ROSING, K. E. and REVELLE, C. S. "Heuristic Concentration: Two Stage solution Construction". *European Journal of Operational Research*. Vol. 97, pp 75-86. (1997).
5. MOSCATO, P.: "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms". *Caltech Concurrent Computation Program, C3P Report 826*. (1989).
6. GLOVER, F.: "Tabu Search: Part I." *ORSA Journal on Computing*. Vol. 1, pp.190-206. (1.989).
7. GLOVER, F.: "Tabu Search: Part II." *ORSA Journal on Computing*. Vol. 2, pp. 4-32. (1990).
8. LAGUNA, M.: "Scatter Search". Aparecerá en *Handbook of Applied (1999). Optimization*, P. M. Pardalos and M. G. C. Resende (eds). Oxford Academic Press.
9. KIRPATRICK S., GELATT C. D. and VECCHI M. P.: "Optimization by Simulated Annealing". *IBM Research Report RC 9355*. (1982).
10. KIRPATRICK S., GELATT C. D. and VECCHI M. P.: "Optimization by Simulated Annealing". *Science*, Vol. 220, pp 671-680. (1983)
11. A. A. MARTÍNEZ MORALES.: "Algoritmo Basado en Tabu Search para el Cálculo del Índice de Transmisión de un Grafo". Departamento de computación Facultad de ciencias y tecnología. FARAUTE de Ciencias y Tecnología, Vol. 1, No. 1, páginas 31-39. Universidad de Carabobo, Valencia, Estado Carabobo, Venezuela (2006).
12. PAOLO TOTH and DANIELLE VIGO.: *The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications*. Society of Industrial and Applied Mathematics. Philadelphia. USA.(2001).
13. WHITLEY D.: "A genetic Algorithm Tutorial", Tech. Report CS-93-103, Colorado State University. (1993).
14. OR, I.: "Traveling Salesman Type Combinatorial Problems and their Relations to the Logistics of Blood Banking." Ph. Thesis. Dpt. of Industrial Engineering y Management Sciences, Northwestern Univ. (1.976).
15. LIN, S.: "Computer Solutions to the Traveling Salesman Problem". *Bell Syst. Tech. Jou.* Vol. 44, pp 2245-2269. (1965).
16. LIN, S. y KERNIGHAN, B. W.: "An Effective Heuristic Algorithm for the Traveling Salesman Problem". *Operations Research*. Vol. 20, pp 498-516. (1973).
17. PACHECO, J. y DELGADO, C.: "Resultados de Diferentes Experiencias con Búsqueda Local Aplicadas a Problemas de Rutas". *Revista Electrónica Rect@. ASEPUMA*, vol.2, nº 1, pp. 54- 81. (2000).
18. GAREY, M. R., JOHNSON, D.S.: *Computers and intractability, A Guide to the theory of NP-Completeness*. W.H.Freeman and Company, New York. USA.ed. (2003).