

Un algoritmo heurístico voraz estocástico para el problema de orientación

Madhushi Verma*

Departamento de Ciencias de la Computación e Ingeniería

Instituto Tecnológico de Massachusetts (BHU)

Benarés, India

madhushi.rs.cse@itbhu.ac.in

(*Autor correspondiente)

Mukul Gupta

Departamento de Ciencias de la Computación e Ingeniería

Instituto Tecnológico de Massachusetts (BHU)

Benarés, India

mukul.gupta.cse10@itbhu.ac.in

Bijeeta Pal

Departamento de Ciencias de la Computación e Ingeniería

Instituto Tecnológico de Massachusetts (BHU)

Benarés, India

bijeeta.pal.cse10@itbhu.ac.in

KK Shukla

Departamento de Ciencias de la Computación e Ingeniería

Instituto Tecnológico de Massachusetts (BHU)

Benarés, India

kkshukla.cse@itbhu.ac.in

Abstracto-En este artículo presentamos un nuevo algoritmo heurístico estocástico voraz para el problema de orientación (OP), que es un problema de grafos de optimización combinatoria NP-Hard. El objetivo del OP es determinar una ruta hamiltoniana que conecte la fuente y el destino especificados, incluya un conjunto de puntos de control y logre la mejor puntuación total posible recopilada dentro del marco de tiempo fijo. Este problema encuentra aplicación en varios campos como la logística, las redes de telecomunicaciones, la industria del turismo, etc. En cada una de estas aplicaciones, es necesario proporcionar un modelo práctico y la mejor manera de abordar esta situación es utilizar grafos incompletos. Sin embargo, la mayoría de las técnicas disponibles en la literatura solo se pueden aplicar en grafos completos. A diferencia de otros algoritmos, nuestro algoritmo se puede aplicar tanto en grafos completos como incompletos. Hemos implementado nuestro algoritmo utilizando cuatro procedimientos de selección diferentes y evaluamos su rendimiento en puntos de referencia estándar. Descubrimos que el algoritmo funciona mejor con la selección de la ruleta.

Palabras clave: Problema de orientación, métodos de selección, gráficos incompletos.

Yo. YoINTRODUCCIÓN

Problema de Orientación (OP), que es una combinación de dos problemas bien conocidos, es decir, el problema del viajante de comercio (TSP) y el problema de la mochila (KP), se originó a partir de un deporte acuático donde el objetivo es determinar una ruta que conecte la fuente y el destino y al mismo tiempo explorar un conjunto de puntos de control. No todos los puntos de control se pueden visitar debido al límite de tiempo disponible (o distancia total recorrida) [1]. OP es un problema de optimización combinatoria NP-Hard y encuentra aplicación en varios campos como la industria del turismo, la planificación de rutas de robots en la gestión de desastres, el sistema de entrega a domicilio, las redes de telecomunicaciones, la logística, las redes de transporte, etc. Varias variantes de OP como la orientación en equipo, la orientación y la orientación en equipo con ventana de tiempo, la orientación generalizada se definen en la literatura que modela diferentes situaciones de la vida real [1].

En 1990, Laporte et al. sugirieron un algoritmo exacto para OP utilizando el concepto de programación lineal [2]. Hayes et al. introdujeron otro algoritmo exacto basado en el

concepto de programación dinámica [3]. Sin embargo, como OP es NP-Hard, no es factible en términos de tiempo de ejecución implementar algoritmos exactos para instancias grandes. Entonces, la técnica eficiente para resolver OP es usar alguna heurística o algoritmo de aproximación y la primera heurística para OP fue propuesta por Tsiligirides en 1984 [4]. En 1991, Ramesh y Brown sugirieron una heurística de cuatro fases [5]. Golden et al introdujeron la heurística del centro de gravedad en 1987 [6]. Fischetti et al presentaron un método de ramificación y corte bien conocido para resolver OP [7]. Más tarde, se propusieron métodos como la búsqueda tabú y las redes neuronales artificiales para resolver OP [8-9]. En 2009, Schilde et al sugirieron técnicas de optimización de colonias de hormigas de Pareto y búsqueda de vecindad de variables de Pareto para tratar con OP [10]. Campos et al en 2013, introdujeron un método llamado Greedy Randomized Adaptive Search Procedure (GRASP) que es una heurística eficiente para abordar el problema de orientación [11]. Para la variante dependiente del tiempo de OP, Fomin et al [12] sugirieron un algoritmo de aproximación. Otros algoritmos de aproximación para la versión no enraizada y la versión enraizada de OP fueron propuestos por Awerbuch et al., Johnson et al y Blum et al respectivamente [13-15]. Como se dijo anteriormente, OP encuentra varias aplicaciones en la vida real y la mayoría de las situaciones prácticas no se pueden modelar solo a través de grafos completos. Se puede presentar una imagen más realista modelando tales situaciones utilizando grafos incompletos. Sin embargo, la mayoría de los algoritmos disponibles en la literatura para OP se pueden aplicar solo en grafos completos. Tasgetiren propuso el primer algoritmo genético para OP, pero en 2011, Ostrowski et al presentaron otro algoritmo genético que se puede aplicar tanto en grafos completos como incompletos [16-17].

En este artículo, presentamos una heurística para OP (SEL_OP) que se puede implementar tanto en grafos completos como incompletos. En nuestro algoritmo, utilizamos diferentes procedimientos de selección y comparamos los resultados para determinar el método de selección que ayuda a lograr la mejor puntuación posible para el problema de orientación.

II. PRE-REQUISITOS

A. Definición del problema

En OP el objetivo es determinar un camino hamiltoniano que conecte la fuente indicada...y objetivo... incluye un subconjunto del conjunto de vértices de modo que la puntuación total recopilada se pueda maximizar dentro del presupuesto de tiempo establecido

OP puede representarse mediante un peso no dirigido grafo (completo o incompleto), donde es el conjunto de vértices y es el conjunto de aristas. Asociamos dos funciones, es decir, una función de tiempo y una función de puntuación a las aristas y vértices respectivamente. Sea V denotar el conjunto de vértices y E denotar el conjunto de aristas. Entonces, $t: V \rightarrow \mathbb{R}^+$ y $p: E \rightarrow \mathbb{R}^+$ significan la función de tiempo y la función de puntuación. un subconjunto $S \subseteq V$. Entonces, $y = \sum_{i \in S} t_i$, tenemos $z = \sum_{e \in E} p_e$ respectivamente [1].

El OP puede plantearse como un problema de programación entera que es el siguiente [1]:

$$\begin{aligned} \sum_{i \in V} x_i &= 1, \quad \sum_{j \in V} x_j = 1 \\ \sum_{i \in V} x_i &\leq 1, \quad \forall i = 2, \dots, n-1 \\ \sum_{j \in V} x_j &\leq 1, \quad \forall j = 2, \dots, n-1 \\ \sum_{i \in V} x_i &\leq 2 \\ 2 \leq x_i &\leq 2, \quad \forall i = 2, \dots, n-1 \\ -1 \leq x_i &\leq 1, \quad \forall i = 2, \dots, n-1 \\ 0, 1 \leq x_i &\leq 1, \quad \forall i = 1, \dots, n \end{aligned}$$

La ecuación 1 representa la función objetivo de OP, es decir, la maximización de la puntuación total obtenida. Algunas otras restricciones, como una ruta, deberían tener: como su inicio y -- Como su nodo final está controlado por la ecuación 2 y ningún vértice se visita más de una vez y la ruta permanece conectada está asegurada por la ecuación 3-4. La condición importante de que la ruta cumpla con el límite de tiempo -- está representado por ecuación 5. La necesidad de eliminar sub tours se ejecuta mediante las ecuaciones 6 y 7. Variable x_i significa la posición de = vértice i en el camino y $x_i = 1$ si i se visita después de i de lo contrario, $x_i = 0$.

B. Métodos de selección

1) Selección del torneo

En esta técnica de selección, se elige un individuo de una población de individuos. Se realizan varios torneos entre los individuos seleccionados aleatoriamente de una población y los mejores se copian a la siguiente generación (aquellos que ganan el torneo, es decir, tienen el mejor valor de aptitud). El proceso se realiza repetidamente varias veces para saturar la población. El tamaño del torneo se denota por q y el tamaño de torneo más común es $q = 2$. Al alterar el tamaño del torneo, la presión de selección se puede ajustar fácilmente. En caso de que el tamaño del torneo sea grande, los individuos débiles tienen menos posibilidades de ser seleccionados. Algunas ventajas de este método son (1) no requiere ningún mecanismo de clasificación y, por lo tanto, se puede implementar en tiempo $O(n)$ y (2) admite una arquitectura paralela [18-20].

2) - , Selección

Esta técnica de selección se formuló con la idea de reducir la población de crías generada como resultado de la recombinación y la mutación. En este caso, la población de crías de tamaño n se reduce seleccionando a los mejores descendientes como nuevos padres para la próxima generación. Es una técnica fácil de implementar, requiere menos tiempo y prácticamente sigue el enfoque codicioso, por lo que genera buenos resultados y ayuda en la mayoría de los casos [18-20].

3) Selección de la ruleta

Este método de selección también se denomina método de selección proporcional y utiliza el enfoque proporcional de aptitud. Selecciona un elemento aleatoriamente de una lista en función de la función de aptitud. Los elementos que tienen un valor de aptitud más alto tienen una mayor probabilidad de ser seleccionados.

- (1) Sin embargo, incluso los elementos con un valor de aptitud menor tienen una probabilidad distinta de cero de ser seleccionados [18-20].

(3)

(4)

(5)

4) Selección aleatoria

Es un método de selección no probabilístico y el más simple. En esta técnica, se selecciona aleatoriamente cualquier nodo.

- (6) del conjunto de nodos candidatos disponibles, es decir, un nodo es seleccionado sólo porque está fácilmente disponible y puede ser

(8) se utiliza convenientemente para realizar operaciones posteriores sin realizar ninguna operación.

cualquier otro procesamiento. Este método puede explorar un gran espacio de búsqueda y proporciona una amplia gama de respuestas. Además, si se permite que la técnica se ejecute una gran cantidad de veces, puede generar la mejor solución posible [18-20].

III. ALGORITMO

Aporte: Un gráfico $G = (V, E)$, con t_i (tiempo que tarda en recorrerlo) valor y valor (Puntaje) de cada arista y de cada vértice respectivamente. Una constante h !valor que denota el número máximo de rutas que se consideran en cada nivel.

Producción: Una ruta hamiltoniana con la mejor puntuación total recopilada posible, de modo que el tiempo total de viaje esté dentro del límite de tiempo establecido.

```

1. # $ % & ' % - h ;
// Matriz de rutas que inicialmente está vacía.
2. h ← ∅;
3. h P ← ( ) -- , - ;
// Ruta más corta entre la fuente ---y objetivo---.
4. * + , % $ ' - h ;
5. $ % ' - $ + . ) . - - h ;

. ) . - - h
1. # $ % & ' % / - h , 0 h 1 2 3 h ;
// Colas que almacenan rutas.
/ - h ← ∅;
0 h 1 2 - h ← ∅;
2. 4 5 $ ← 0 | - h |

```

```

1 7 . -- h 8 9, 0h12-h ;
//h h contendrá niños generados a partir de
Cada camino en h .
3.45$ ← 0 - h !
//Seleccionando mejor h niños Para el próximo
generación.
.0h12- h =: - h-0h12-h ;
//La ruta con la puntuación total máxima.
;%<5=% 0h12-h )>0h12-h ;
* + , %$' 0h12- ? . h /-h ;
@4 | 0h12- h | ==0
?) ;
4. @4 /-h == - h
// Terminar si no se genera un nuevo hijo y devolver el
h.
$%- $+ : - h-- h ;
5. $%- $+ .) .- /- h ;

1 7 . --h -, 0h12- h
// hes una matriz de nodos que forma una ruta.
1.A ← 0;
2.45$ ← 0 | |
3. . @4€ -
// Si ya hay un nodo presente en la ruta, ignórelo.
B5+'@+-%;
?. #&CB-C&'%Δ;
// El incremento de tiempo debido a la inserción de -En su mejor posición.
/Δ | , Δ ≥ 1
7. A = D , - 1 ≤ Δ < 1
* |Δ | , Δ < -1

2. @4A ≤A
A ← A
4.0) 0,2 2 ;
// Matriz de nodos candidatos utilizados para la selección.
5.0,2 2 ← Ø;
6.45$ ← 0 | |
7. . @4€ -
B5+'@+-%;
?. @4 (Un≥FA ) && ( + Δ ≤ )
// es el parámetro de codicia que decide qué nodo debe participar
en el proceso de selección.
7. *+, %$' - 0,2 2 ;
8. @4 | 0.2 2 | ==0
h h ;
// Si no se agregan nuevos nodos, inserte la ruta principal .
;%'- $+;
9.45$ ← 0 - - h - -
// Genera h Número de rutas secundarias de la ruta P.
10. .0h12 2←0 >3 - 0,2 2

// Calcular 1 usando la selección del
torneo. // Calcular 2 usando la selección ( , ).
// Calcule 3 usando la selección de la ruleta. //
Calcule 4 usando la selección aleatoria.

```

```

;%<5=% 0h12 2 )>0,2 2 ;
* + , %$' 0h12 ; %<5=% 0h12h - . 2 *+, %$' - h - 0h12- h .
2G> - h - ;
?. @4 | 0,2 2 | =0
M$%&I;

: - h-- h
1. > ← 0; ? 3 h ← 0;
2.45$ ← 0 | -h |
a. @4 | 7 ) - h - K >> 7 )
//Puntaje ( h ) es la suma de las recompensas asociadas a cada
nodo de h .
?. > 7 ) ← 7 ) - h - ;
? 3 h ← ;
3. ) ) .- h - ? 3h ;

```

El objetivo del algoritmo mencionado anteriormente es determinar una ruta que conecte la fuente: y objetivo--y un subconjunto de vértices que maximizan la puntuación total recopilada y obedecen el límite de tiempo. Este algoritmo se puede implementar tanto en gráficos completos como incompletos. Para garantizar que en una ruta, la fuente y el destino estén conectados, implementamos el algoritmo de Dijkstra como se muestra en el paso 3 de _"-(). A

Para tener en cuenta la otra restricción de que ningún vértice se visita más de una vez, los nodos explorados se eliminan del conjunto de nodos disponibles que forman la lista de candidatos para el proceso de selección. 1 7 . () , cuatro selecciones

Se utilizan técnicas y en la siguiente sección se presenta una comparación de sus resultados. La complejidad temporal de _"-es "-| | .

IV. EXPERIMENTAL AANÁLISIS

Nuestro código se implementó en C++ y se compiló utilizando CodeBlocks en un Intel Core i5 650 funcionando a 2,20 GHz.

_"-tiene la capacidad de abordar tanto gráficos completos como incompletos y aquí informamos los resultados para gráficos incompletos ejecutando el código en algunos datos reales. Consideramos dos casos, es decir, datos reales de la red de carreteras de 160 y 306 ciudades de Polonia. Cada caso está asociado con dos archivos, es decir, cities.txt y distances.txt. Los nombres de las distintas ciudades y sus puntuaciones se especifican en cities.txt. La puntuación de cada ciudad se calcula en función del número de habitantes utilizando la fórmula $7) = . h ? /10000$. El otro archivo distances.txt representa para cada ciudad, su ciudad adyacente y sus respectivas longitudes de aristas. Estas aristas del gráfico corresponden a las carreteras del mapa real de Polonia.

El algoritmo _"-Se ejecutó utilizando cuatro diferentes métodos de selección definidos en la sección II (B). En la Tabla 1, el valor máximo y medio de la puntuación obtenida por cada técnica de selección para diferentes Los valores son

En la Figura 1 se presentan estos valores para la primera instancia con 160 ciudades. Se observó que el método de selección de ruleta funciona mejor que los otros tres.

técnicas en la mayoría de los casos y ayuda a obtener una mejor puntuación total recopilada en comparación con las otras técnicas.

También se observó que cuando se ejecutó el mismo algoritmo para la instancia de 306 ciudades, solo el método de selección aleatoria sufre y el resto de los métodos de selección logran el mismo puntaje total máximo recopilado la mayoría de las veces, como se muestra en la Tabla 2 y la Figura 2. La posible razón de esta observación podría ser la existencia de grandes grupos en la instancia de 306 ciudades, lo que está ausente en el caso de la instancia de 160 ciudades, como se muestra en la Figura 3 (a) y 3 (b). Por lo tanto, las técnicas de selección tienen más opciones posibles disponibles para ser exploradas en el caso de la instancia de 160 ciudades y no se quedan estancadas dentro de un grupo. Por lo tanto, se puede afirmar aquí que el procedimiento de selección de ruleta funciona mejor que los otros métodos cuando se ejecuta por el algoritmo.

V.C.CONCLUSIÓN

En este artículo, presentamos un nuevo enfoque heurístico estocástico voraz (el algoritmo) para la orientación Problema que se puede aplicar tanto a gráficos completos como incompletos. Se implementó utilizando cuatro métodos de selección diferentes y los resultados así obtenidos se compararon en puntos de referencia estándar. Descubrimos que el método de selección de ruleta funciona mejor que las otras tres técnicas de selección y ayuda a lograr una mejor puntuación total recopilada para el presupuesto de tiempo especificado.

AAGRADECIMIENTO

El primer autor desea agradecer el apoyo financiero del IIT (BHU) en términos de ayudantía de docencia.

RREFERENCIAS

- [1]P. Vansteenwegen, W. Souffriau y DV Oudheusden, "El problema de la orientación: una encuesta", *European Journal of Operational Research*, vol. 209, págs. 1-10, 2011.
- [2]G. Laporte y S. Martello, "El problema del viajante selectivo", *Discrete Applied Mathematics*, vol. 26, págs. 193-207, 1990.
- [3]M. Hayes y JM Norman, "Programación dinámica en orientación: elección de ruta y ubicación de los controles", *Journal of the Operational Research Society*, vol. 35 (9), págs. 791-796, 1984.
- [4]T. Tsiligirides, "Métodos heurísticos aplicados a la orientación", *Journal of the Operational Research Society*, vol. 35, págs. 797-809, 1984.
- [5]R. Ramesh y K. Brown, "Una heurística eficiente de cuatro fases para el problema de orientación generalizado", *Computers and Operations Research*, vol. 18, págs. 151-165, 1991.
- [6]B. Golden, L. Levy y R. Vohra, "El problema de la orientación", *Naval Research Logistics*, vol. 34, págs. 307-318, 1987.
- [7]M. Fischetti, J. Salazar y P. Toth, "Resolución del problema de orientación mediante ramificaciones y cortes", *INFORMS Journal on Computing*, vol. 10, págs. 133-148, 1998.
- [8]Q. Wang, X. Sun, BL Golden y J. Jia, "Uso de redes neuronales artificiales para resolver el problema de orientación", *Annals of Operations Research*, vol. 61, págs. 111-120, 1995.
- [9]M. Gendreau, G. Laporte y F. Semet, "Una heurística de búsqueda tabú para el problema del viajante de comercio selectivo no dirigido", *European Journal of Operational Research*, vol. 106, págs. 539-545, 1998.
- [10]M. Schilde, KF Doerner, RF Hartl y G. Kiechle, G. "Metaheurísticas para el problema de orientación bi-objetivo", *Swarm Intelligence*, vol. 3, págs. 179-201, 2009.
- [11]V. Campos, R. Marti, J. Sanchez-Oro y A. Duarte, "GRASP con Path Relinking para el problema de orientación", *Journal of the Operational Research Society* (2013), págs. 1-14, 2013.
- [12]FV Fomin y A. Lingas, "Algoritmos de aproximación para orientación dependiente del tiempo", *Information Processing Letters*, vol. 83, págs. 57-62, 2002.
- [13]B. Awerbuch, Y. Azar, A. Blum y S. Vempala, "Garantías de aproximación mejoradas para árboles k de peso mínimo y vendedores coleccionistas de premios", *Siam J. Computing*, vol. 28, págs. 254-262, 1999.
- [14]D. Johnson, M. Minkoff y S. Phillips, "El problema del árbol de Steiner para la recolección de premios: teoría y práctica", *Actas del 11º Simposio Anual ACM-SIAM sobre Algoritmos Discretos*, 2000, págs. 760-769.
- [15]A. Blum, S. Chawla, DR Karger, T. Lane, A. Meyerson y M. Minkoff, "Algoritmos de aproximación para orientación y TSP con recompensa descontada", *Actas del 44º Simposio anual IEEE sobre fundamentos de la informática (FOCS'03)*, 2003, págs. 1-10.
- [16]M. Tasgetiren, "Un algoritmo genético con una función de penalización adaptativa para el problema de orientación", *Journal of Economic and Social Research*, vol. 4(2), págs. 1-26, 2001.
- [17]K. Ostrowski y J. Koszelew, "La comparación de algoritmos genéticos que resuelven problemas de orientación utilizando gráficos completos e incompletos", *Informatyka*, vol. 8, págs. 61-77, 2011.
- [18]T. Back, "Presión selectiva en algoritmos evolutivos: una caracterización de los mecanismos de selección", *Actas de la Primera Conferencia IEEE sobre Computación Evolutiva*, 1994, págs. 57-62.
- [19]NM Razali y J. Geraghty, "Rendimiento del algoritmo genético con diferentes estrategias de selección para resolver TSP", *Actas del Congreso Mundial de Ingeniería 2011*, vol II WCE 2011, 6 al 8 de julio de 2011, Londres, Reino Unido, págs. 1134-1139.
- [20]R. Sivaraj y T. Ravichandran, "Una revisión de los métodos de selección en algoritmos genéticos", *Revista internacional de ingeniería, ciencia y tecnología (IJEST)*, vol. 3(5), págs. 3792-3797, 2011.

TABLA I. Comparación del valor medio y máximo de la puntuación total recogida obtenida por ---_ cuando se ejecuta con cuatro selecciones diferentes
procedimientos para 160 ciudades.

	(μ, λ)		Aleatorio		Torneo		Rueda de la ruleta	
	Máximo	Significar	Máximo	Significar	Máximo	Significar	Máximo	Significar
500	49	49	49	48	49	48	49	48
750	67	67	67	64	67	66	67	65
1000	83	83	88	77	88	84	88	79
1250	103	103	103	99	103	101	102	100
1500	118	118	116	114	119	116	116	114
1750	135	135	130	128	135	128	130	128
2000	145	145	145	143	145	143	148	143
2250	157	157	160	154	157	155	162	155
2500	179	179	184	175	179	176	187	174
2750	190	190	203	193	191	189	203	193
3000	206	206	214	205	206	202	214	206
3250	234	234	233	223	234	228	233	224
3500	248	248	249	242	249	245	251	244
3750	255	255	261	255	257	255	263	256
4000	269	269	271	264	272	268	270	267

TABLA II. Comparación del valor medio y máximo de la puntuación total recogida obtenida por ---_ cuando se ejecuta con cuatro selecciones diferentes
procedimientos para 306 ciudades.

	(μ, λ)		Aleatorio		Torneo		Rueda de la ruleta	
	Máximo	Significar	Máximo	Significar	Máximo	Significar	Máximo	Significar
500	127	121	127	117	127	121	127	118
750	155	149	155	147	155	148	155	147
1000	178	173	177	168	178	172	178	169
1250	211	173	208	190	211	172	211	193
1500	236	213	234	218	236	219	236	221
1750	263	241	263	253	263	254	263	251
2000	282	280	279	273	282	277	282	273
2250	307	300	299	291	307	292	307	281
2500	325	323	323	307	325	309	325	306
2750	349	336	333	325	349	327	349	329
3000	366	355	366	346	366	345	366	344
3250	396	339	386	366	396	371	396	369
3500	420	394	414	384	420	393	420	390
3750	435	408	428	410	435	411	435	400
4000	459	430	448	429	459	437	459	440

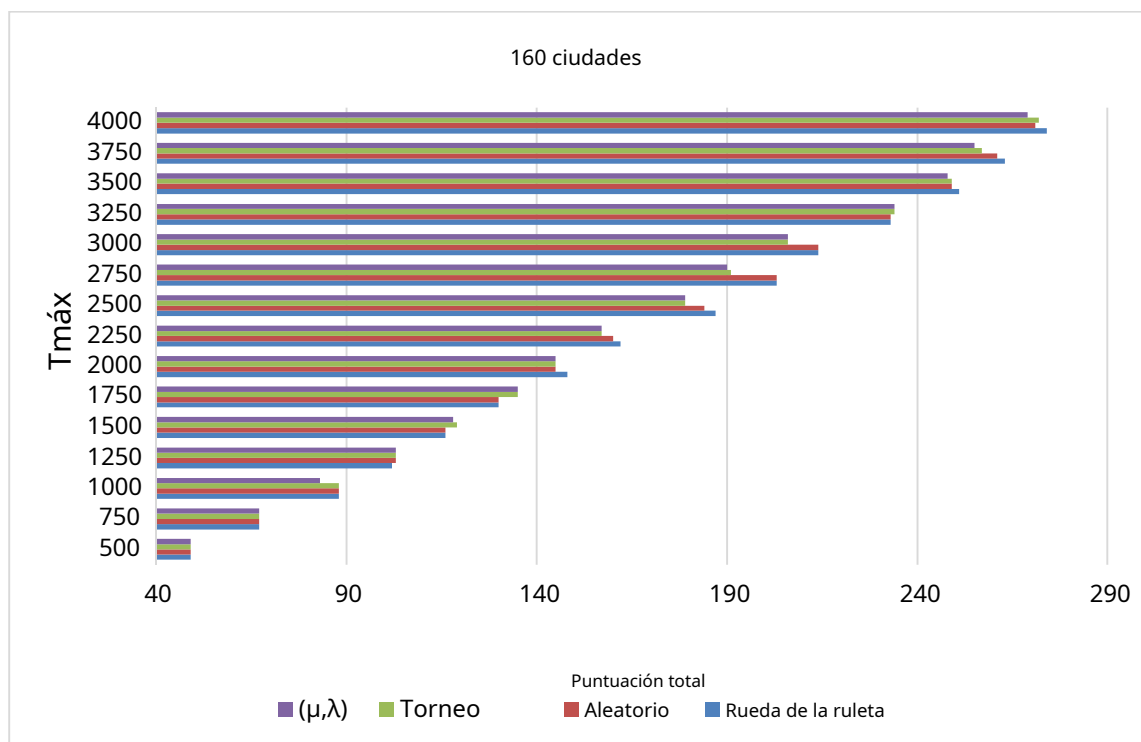


Figura 1. Comparación del valor máximo de la puntuación total recopilada obtenida por cuatro métodos de selección diferentes para diferentes...valores (160 ciudades).

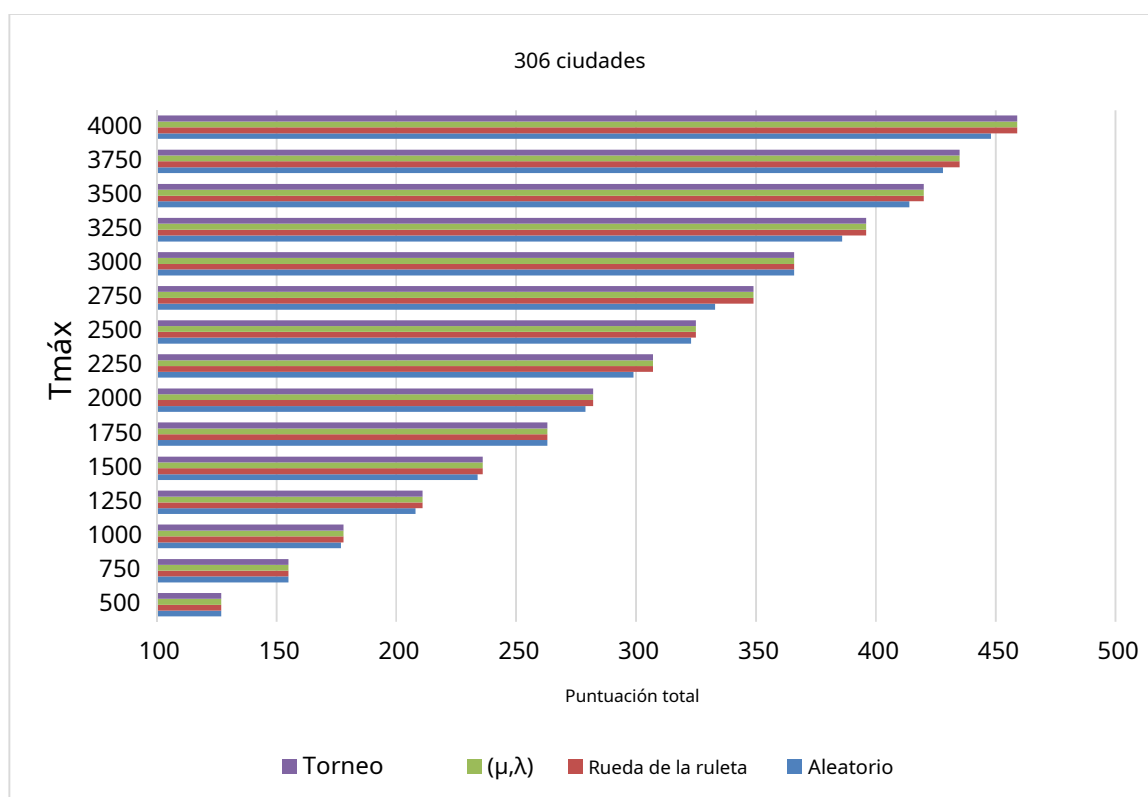
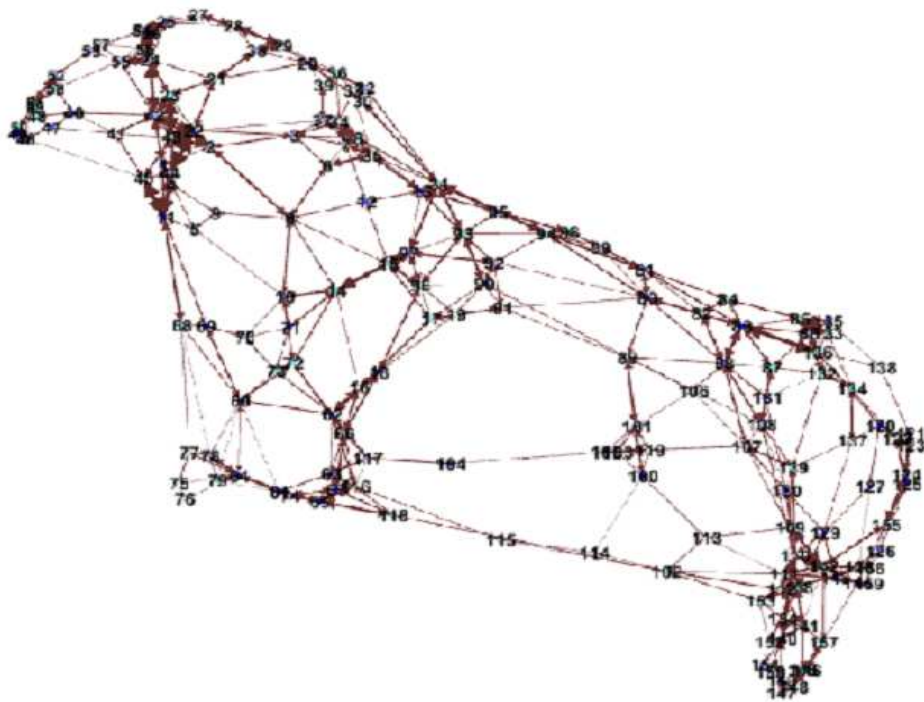
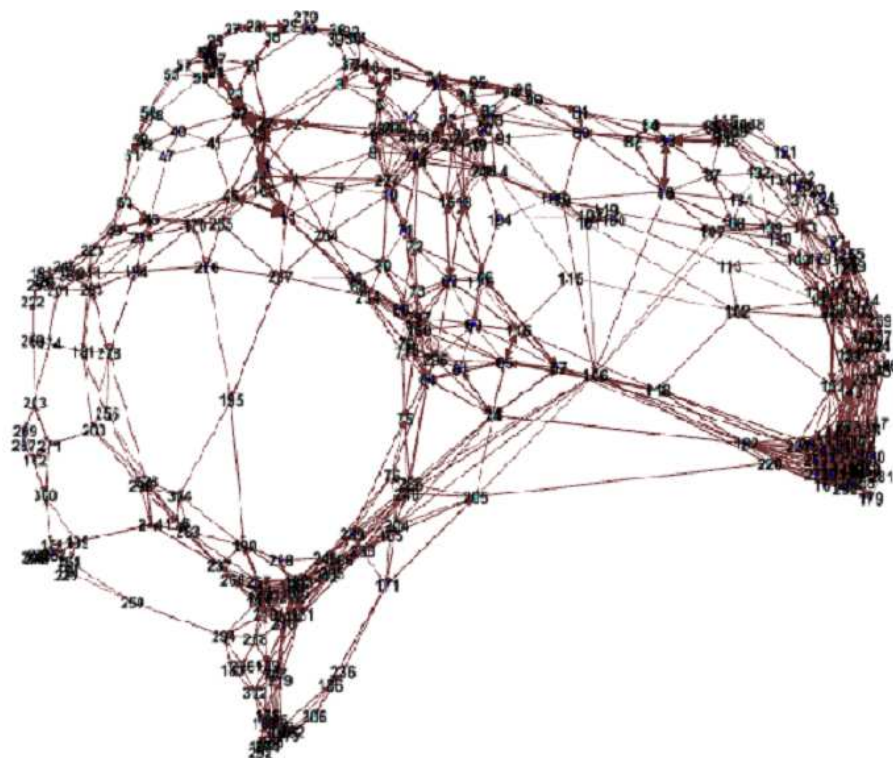


Figura 2. Comparación del valor máximo de la puntuación total recopilada obtenida por cuatro métodos de selección diferentes para diferentes...valores (306 ciudades).



(a)



(b)

Figura 3. Gráfico para instancias de (a) 160 ciudades y (b) 306 ciudades.