Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 25/05/2021 | Aceptado: 11/10/2021

Determinación experimental de la influencia de la reformulación del problema RAP en su eficiencia computacional

Experimental determination of the influence of the reformulation of the RAP problem on its computational efficiency

Randy Reyna-Hernández [1*] https://orcid.org/0000-0003-1481-9546
Alejandro Rosete [2] https://orcid.org/0000-0002-4579-3556

[1] Universidad de Matanzas. Carretera a Varadero, Km 3½, Matanzas, Cuba. 40100. randyrh91@gmail.com
[2] Universidad Tecnológica de La Habana José Antonio Echeverría. Cujae, Marianao. 19390. rosete@ceis.cujae.edu.cu

*Autor para correspondencia. (randyrh91@gmail.com)

RESUMEN

El problema de agregación de rankings (RAP, por sus siglas en inglés) busca encontrar un ranking que resuma un conjunto de ellos. Actualmente, el razonamiento sobre la base de ordenamientos o rankings ha ganado atención debido al gran número de aplicaciones para solucionar problemas de toma de decisiones y más recientemente, en informática, estadística, algebra lineal y optimización, la biología computacional entre muchas otras. Dentro del contexto de RAP se encuentra el Problema del Ranking de Kemeny (KRP), donde todos los rankings de entrada son una permutación. El KRP es NP-duro, sin embargo, existe una

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

formulación donde, a través del método de Programación Lineal Entera (PLE) se le puede dar solución al problema. Como la eficiencia del método de PLE está dada por la cantidad de variables y restricciones, se obtuvo una reformulación del problema que permite reducir ambas dimensiones. En el presente trabajo se demuestra como la reformulación del RAP permite resolver con el mismo software instancias mayores y de resolver las mismas en menos tiempo.

Palabras clave: Agregación de Rankings; Programación Lineal Entera; Plugins de KNIME; Problema del Ranking de Kemeny.

ABSTRACT

The Rank Aggregation Problem (RAP) search for a ranking that summarizes a set of them. Currently, the reasoning based on rankings has gained great attention due to the large number of applications to solve decision-making problems and more recently, in computer science, statistics, linear algebra and optimization, computational biology among many others. A particular case of RAP is the Kemeny Ranking Problem (KRP), where all entry rankings are a permutation. The KRP is NP-hard, however, there is a formulation where, through the method of Integer Linear Programming (PLE), the problem can be solved. As the efficiency of the PLE method is given by the number of variables and restrictions, a reformulation of the problem was obtained that allows both dimensions to be reduced. In the present work it is demonstrated how the reformulation of the RAP allows to solve with the same software major instances and to solve them in less time.

Keywords: Ranking Aggregation; Integer Linear Programming; KNIME Plugins; Kemeny Ranking Problem.

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

Introducción

El Problema de Agregación de Rankings (RAP) es el proceso de combinar múltiples listas de rankings (anglicismo aceptado por la Real Academia de la Lengua Española), denominados "rankings base" en un solo ranking ordenado, denominado como "ranking agregado" que tiende a ser más confiable que los rankings base (Liu, 2019).

En los últimos años el razonamiento sobre la base de ordenamientos o rankings ha ganado gran atención debido a sus disímiles aplicaciones para solucionar problemas de toma de decisiones (Tao, 2019) y más recientemente, bioinformática (Galdi, 2019; Liu, 2019), metabúsquedas (Galdi, 2019), procesamiento del lenguaje natural (Onan & Korukoğlu, 2017), búsquedas web (Kaur, 2021), bibliometría (Subochev, 2018), entre muchas otras.

Para resolver el RAP se han planteado numerosas soluciones, mediante algoritmos como: Ramas y Cotas, (Ali & Meila, 2012) Método de Borda (Xiao, 2017), Método de Copeland (Lestari, 2018), algoritmos genéticos (Aledo, 2018) o First Order Then Append (FOTA) (Aledo, 2021).

Además, se ha estudiado la posibilidad de usar métodos exactos para su resolución (Ali & Meila, 2012), sobre todo, si se puede reducir la dimensión del problema sin afectar la solución, debido a que las dimensiones de estos problemas tienen una influencia directa en el tiempo de ejecución y los recursos necesarios para resolverlos. En este sentido, en (Rosete-Suárez, 2018) se presenta una reformulación del RAP para la solución a través de PLE que permite reducir de manera notable, tanto las variables como las restricciones del problema. Esto no cambia el orden de complejidad temporal del problema, pero crea posibilidades para enfrentar instancias mayores, con ahorro de recursos computacionales.

Por tanto, este trabajo se centra en demostrar cómo se reducen las variables a la mitad y restricciones a un tercio, aproximadamente, así como la influencia directa de la reducción de las variables y las restricciones en el tiempo de ejecución y los recursos necesarios para resolver el problema, a partir de la reformulación del problema de PLE planteada en (Rosete-Suárez, 2018).

El resto del documento está estructurado de la siguiente manera. A continuación, se define el RAP, además de la formulación de PLE para el RAP y la reformulación planteada en (Rosete-Suárez, 2018). Posteriormente, se presentan los resultados de los experimentos realizados que permiten evaluar la mejora

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

en la eficiencia computacional de la solución del RAP. Por último, se dan a conocer las conclusiones del presente trabajo.

Métodos

El Problema de Agregación de Ranking (RAP, por sus siglas en inglés) tiene el objetivo de encontrar una permutación que minimiza la distancia respecto a un conjunto de rankings que se reciben como entrada (Chatterjee, 2018). En el RAP es usual representar el conjunto de rankings de entrada en una matriz que los resume a todos, comúnmente llamada Matriz de Precedencia (Aledo, 2021).

Problema de Agregación de Ranking (RAP)

Partiendo del hecho que existe una matriz P (Matriz de Precedencia), el RAP pueden ser resueltos a través del método de Programación Lineal Entera (PLE), produciendo una matriz binaria X, donde cada celda X_{uv} de la matriz X indica la relación de precedencia entre el elemento "u" y el elemento "v". Si celda $X_{uv} = 1$ entonces "u" precede a "v" y 0 en el caso contrario. Entonces, el objetivo del problema RAP es (Rosete-Suárez, 2018):

$$minimizar\left(\sum_{u,v} P_{uv}x_{vu} + P_{vu}x_{uv}\right),$$

Sujeto a tres restricciones:

- Restricción 1: $x_{uv} \in \{0,1\}, \forall_{u,v}$
- Restricción 2: $x_{uv} + x_{vu} = 1, \forall_{u,v}$
- Restricción 3: $x_{uv} + x_{vt} + x_{tu} \geq 1, \forall_{u.v,t}$

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

Reformulación del Problema de Agregación de Ranking (RAP)

En (Rosete-Suárez, 2018) se plantea una reformulación del RAP para su la solución por PLE, donde se reducen las variables y restricciones del problema original, planteado anteriormente. A continuación, se detalla la reformulación.

Para la reducción de variables es importante notar que realmente cada variable en la Restricción 2 es dependiente del valor de X_{uv}. Esto implica que se puede rescribir esta restricción de la manera siguiente:

- $X_{uv} + X_{vu} = 1, \ \forall u, v$
- $X_{vu} = 1 - X_{uv}, \ \forall u, v$

Esto trae consigo dos implicaciones directas. Primero, se elimina la Restricción 2, porque ya no habría que comprobar su cumplimiento (no es posible incumplirla). Segundo, se reduce a la mitad la cantidad de variables, debido a que basta con representar los casos donde u < v, ya que el resto de casos son calculables. De tal forma que ahora la función objetivo podría transformarse de la forma siguiente:

$$minimizar \ P_s + \sum_{\substack{u,v \\ u<v}} X_{uv} \ D_{uv}$$

Donde:

- P_s es un valor constante que se corresponde con la suma de los elementos del triángulo superior de la matriz P_{uv}
- $D_{uv} = P_{vu} - P_{uv}$

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

Por otro lado, a partir de un análisis en detalle de la Restricción 3, donde se observa que siguiendo el razonamiento que llevó a la reducción de las variables, ahora se podría expresar en función de las variables $X_{uv}, u < v.$

De esta manera, se reducen a dos restricciones de la forma siguiente para cada una de las combinaciones (u, v, t) tales que u < v, < t.

- $X_{uv} + X_{vt} - X_{ut} \geq 0, \text{para } u < v, < t$
- $X_{uv} + X_{vt} - X_{ut} \leq 1, \text{para } u < v, < t$

Diseño experimental

Para comprobar la influencia que tiene la reformulación previamente descrita en la eficiencia de la solución de PLE del RAP, se diseñó un experimento basado en medir la eficiencia computacional de ambas formulaciones en 50 conjuntos de datos (dataset) de rankings reales disponibles en (Mattei & Walsh, 2013). En particular, se descargaron ficheros pwg asociados a los siguientes conjuntos de "Datos de Elección": ED-00006-Skate Data (3–4, 11–12, 18, 28, 46, 48), ED-00011-Web Search (1–2), ED-00014-Sushi Data (1) and ED-00015-Clean Web Search (1–4, 7, 9,12, 14, 16–20, 23–25, 27, 29–30, 32, 34, 40–42, 44, 46, 48, 50, 54, 55,57, 59, 65–66, 67, 69, 73, 74, 77).

Todos los experimentos fueron realizados en un ordenador personal con un procesador Intel i7 - 4790, 3.60 GHz, 4 núcleos y 4GB de memoria RAM.

Una descripción general de estos ficheros se muestra en la tabla 1. Por cada conjunto de datos, se muestra el promedio (Pro), la mediana (Med), los valores mínimos (Min) y máximos (Max) y la desviación estándar (DesEst) del número de elementos del conjunto de rankings a agregar (n) y la cantidad de votantes (v).

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

Tabla 1 - Descripción de los conjuntos de rankings usados en los experimentos.

	Pro.	**Med.**	**Min**	**Max.**	**DesEst.**
n	88.66	70	10	242	69.72
v	104.76	4	4	5000	706.42

Para realizar los experimentos fue necesario implementar varios nodos para la herramienta de minería de datos KNIME (Universidad de Constanza, 2020) que permiten cargar los ficheros PWG y a partir de ellos, generar modelos de PLE para ser ejecutados en dos de las herramientas de software libre que solucionan problemas de optimización (comúnmente llamados "solvers"): LiPS (KONOBEY, 2019) y SCIP (Z. I. Berlin, 2017).

Resultados y discusión

En esta sección se presenta el resultado experimental de las comparaciones entre el Método de Programación Lineal Entera (MPLE) y la Reformulación del Método de Programación Lineal Entera (RMPLE) en cuanto a variables y restricciones. De igual forma, se muestra la factibilidad de resolver con cada uno de los softwares (LiPS y SCIP) los modelos de PLE generados desde los nodos de KNIME desarrollados. Por último, se muestra una comparación entre los tiempos empleados en cada par instancia-solver.

Reducción de variables y restricciones

La tabla 2 muestra para cada fichero PWG una comparación entre la cantidad de variables para MPLE (v_vn) y la cantidad de variables para RMPLE (v_vr). Además, se comparan la cantidad de restricciones para MPLE (r_vn) y la cantidad de restricciones para RMPLE (r_vr).

Tabla 2 - Ventajas de RMPLE.

	n	**v_vn**	**v_vr**	**r_vn**	**r_vr**		**n**	**v_vn**	**v_vr**	**r_vn**	**r_vr**
06_03	14	182	91	2366	728	**15_09**	115	13110	6555	1494540	493810

06_04	14	182	91	2366	728	**15_12**	100	9900	4950	980100	323400
06_11	20	380	190	7220	2280	**15_14**	163	26406	13203	4277772	1417122
06_12	20	380	190	7220	2280	**15_16**	70	4830	2415	333270	109480
06_18	24	552	276	12696	4048	**15_17**	127	16002	8001	2016252	666750
06_28	24	552	276	12696	4048	**15_18**	115	13110	6555	1494540	493810
06_46	30	870	435	25230	8120	**15_19**	87	7482	3741	643452	211990
06_48	24	552	276	12696	4048	**15_20**	122	14762	7381	1786202	590480
11_01	240	57360	28680	13709040	4550560	**15_23**	142	20022	10011	2823102	934360
11_02	242	58322	29161	14055602	4665760	**15_40**	131	17030	8515	2196870	732290
14_01	10	90	45	810	240	**15_41**	70	4830	2415	328440	109480
15_01	240	57360	28680	13709040	4550560	**15_42**	100	9900	4950	970200	323400
15_02	240	57360	28680	13709040	4550560	**15_44**	45	1980	990	85140	28380
15_03	242	58322	29161	14055602	4665760	**15_46**	40	1560	780	59280	19760
15_24	91	8190	4095	737100	242970	**15_48**	10	90	45	720	240
15_25	115	13110	6555	1481430	493810	**15_50**	26	650	325	15600	5200
15_27	95	8930	4465	830490	276830	**15_54**	60	3540	1770	205320	68440
15_29	106	11130	5565	1157520	385840	**15_55**	52	2652	1326	132600	44200
15_30	64	4032	2016	249984	83328	**15_57**	73	5256	2628	373176	124392
15_32	153	23256	11628	3511656	1170552	**15_59**	55	2970	1485	157410	52470
15_34	55	2970	1485	157410	52470	**15_65**	40	1560	780	59280	19760
15_04	242	58322	29161	14055602	4665760	**15_66**	52	2652	1326	132600	44200
15_07	110	11990	5995	1306910	431640	**15_67**	30	870	435	24360	8120
15_69	81	6480	3240	511920	170640	**15_74**	20	380	190	6840	2280
15_73	36	1260	630	42840	14280	**15_77**	56	3080	1540	166320	55440

Las figura 1 y la figura 2 muestran un gráfico comparativo del crecimiento de la cantidad de variables y restricciones, respectivamente, para MPLE y RMPLE.

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

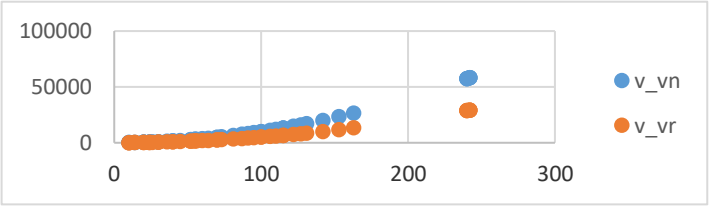

Fig. 1 – Crecimiento de la cantidad de variables en ambas formulaciones.

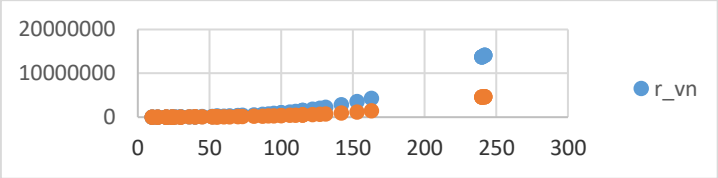

Fig. 2– Crecimiento de la cantidad de restricciones en ambas formulaciones.

La figura 3 muestra un gráfico comparativo, con la proporción de la reducción de variables y restricciones en RMPLE según crece n.

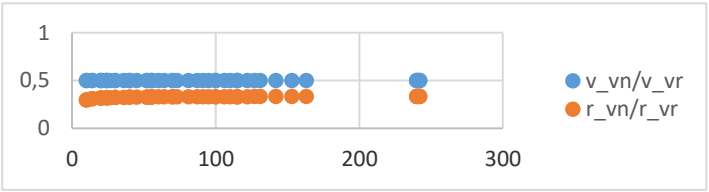

Fig. 3 – Reducción de la cantidad de variables y restricciones con RMPLE.

Como se puede apreciar en la tabla comparativa y en los gráficos, con la RMPLE se logra una reducción de la mitad de variables con respecto al MPLE y aproximadamente un tercio de las restricciones.

Influencia en la factibilidad para resolver cada instancia con LiPS y SCIP

La tabla 3 muestra los valores mínimos (min) y máximo (max) de la cantidad de elementos a ordenar (n), la cantidad de variables (v) y la cantidad de restricciones (r) que se pudieron resolver con LiPS y SCIP

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

respectivamente. Además, se puede apreciar el total de instancias resultas (t) para cada formulación y el porciento (%) que representa del total de instancias.

Tabla 3 - Rango de valores para las instancias resueltas con LiPS y SCIP para MPLE y RMPLE.

	LiPS								**SCIP**							
	n-min	**n-max**	**v-min**	**v-max**	**r-min**	**r-max**	**t**	**%**	**n-min**	**n-max**	**v-min**	**v-max**	**r-min**	**r-max**	**t**	**%**
MPLE	10	10	45	45	240	240	2	4	10	163	45	26406	240	4277772	44	88
RMPLE	10	24	45	276	240	6840	10	20	10	240	45	28680	240	13651680	47	94

Como se puede apreciar en la tabla 3, usando LiPS la reformulación permitió resolver instancias con hasta 24 elementos, mientras que la formulación original solo permitía llega hasta 10. En tanto, con SCIP llegaron a resolver instancias con 240 elementos mientras que la formulación original llegaba hasta 163 elementos.

Influencia en el tiempo para resolver cada instancia con LiPS Y SCIP

La tabla 4 muestra una comparación en cuanto al tiempo de solución (en segundos) de cada una de las instancias con LiPS para MPLE (LiPS_tvn) y RMPLE (LiPS_tvr) y con SCIP para MPLE (SCIP_tvn) y RMPLE (SCIP_tvr). Las celdas con espacios en blanco son resultado de instancias que no se le pudo dar solución con el solver debido a sus dimensiones.

Tabla 4 - Tiempo de solución de cada instancia (en segundos) con LiPS y SCIP para MPLE y RMPLE respectivamente.

	n	**LiPS_tvn**	**LiPS_tvr**	**SCIP_tvn**	**SCIP_tvr**		**n**	**LiPS_tvn**	**LiPS_tvr**	**SCIP_tvn**	**SCIP_tvr**
06_03	14		1,516	0,016	0,007	**15_19**	87			3,566	1,598
06_04	14		0,299	0,017	0,005	**15_20**	122			10,928	4,596
06_11	20		42,945	0,041	0,023	**15_59**	55			0,810	0,372
06_12	20		12,762	0,041	0,017	**15_65**	40			0,306	0,140
06_18	24		54,515	0,063	0,036	**15_66**	52			0,698	0,316
06_28	24		191,140	0,065	0,040	**15_24**	91			4,058	1,788

06_46	30			0,129	0,051	**15_25**	115			9,050	3,802
06_48	24		55,399	0,064	0,030	**15_27**	95			4,732	2,096
11_01	240				61,310	**15_29**	106			6,798	2,904
11_02	242					**15_30**	64			1,318	0,606
14_01	10	17,981	0,063	0,002	0,001	**15_32**	153			27,170	9,652
15_01	240				62,360	**15_34**	55			0,806	0,386
15_02	240				58,890	**15_40**	131			14,314	5,916
15_03	242					**15_41**	70			1,692	0,786
15_04	242					**15_42**	100			5,712	2,514
15_23	142			19,290	7,498	**15_44**	45			0,432	0,204
15_55	52	17,981	17,981	0,694	0,308	**15_46**	40			0,280	0,146
15_57	73			1,986	0,894	**15_48**	10	17,727	0,063	0,025	0,002
15_07	110	17,981	17,981	7,578	3,310	**15_50**	26			0,090	0,034
15_09	115			8,858	3,843	**15_54**	60			1,036	0,498
15_12	100			5,593	2,684	**15_67**	30			0,120	0,046
15_14	163			37,900	11,822	**15_69**	81			2,778	1,306
15_16	70			1,760	0,776	**15_73**	36			0,204	0,094
15_17	127			12,728	5,264	**15_74**	20		58,507	0,038	0,010
15_18	115			8,994	3,846	**15_77**	56			0,850	0,420

Como se puede apreciar en la tabla 4, se pudieron resolver las mismas instancias en un menor tiempo, o sea, las instancias en las nueva formulación se resolvieron empleando menos del 5% del tiempo para LiPS y menos del 50% para SCIP.

Las Figuras 4 y 5 muestran una comparación entre LiPS y SCIP, en cuanto a los tiempos (en segundos) que se demoraron en dar solución a los modelos.

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

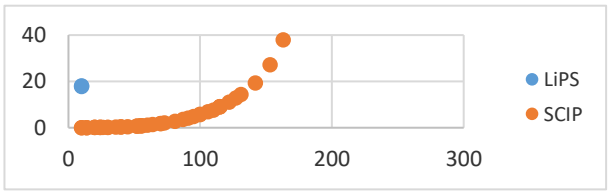

Fig. 4 – Tiempos de solución para MPLE.

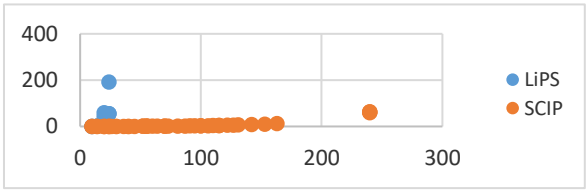

Fig. 5 – Tiempos de solución para RMPLE.

Como se aprecia en la figura 4 y la figura 5, el tiempo de ejecución para RMPLE de los modelos aumenta de manera menos notable que el tiempo de ejecución de los modelos para MPLE. Este comportamiento demuestra la mayor eficiencia de RMPLE con respecto a MPLE para ejecutar modelos con un mayor número de elementos del conjunto de rankings a agregar.

Conclusiones

A partir de los experimentos realizados, se puede observar, en primer lugar, cómo el Método de Programación Lineal Entera permite resolver instancias de hasta 10 elementos del conjunto de rankings a agregar usando LiPS y de hasta 163 usando SCIP, mientras tanto, la Reformulación del Método de Programación Lineal Entera, permite resolver instancias con un notable aumento del número de elementos del conjunto de rankings a agregar, hasta 24 usando LiPS y 240 usando SCIP.

Además, en cuanto al tiempo de ejecución de las instancias, también se puede observar que con la Reformulación del Método de Programación Lineal Entera hay una notable mayor eficiencia de las

herramientas LiPS y SCIP para resolver cada una de las instancias, ya que a medida que aumentan los elementos del conjunto de rankings a agregar hay un notable aumento del tiempo con el Método de Programación Lineal Entera, sin embargo, no se hace tan notable el aumento del tiempo al analizar los resultados obtenidos con la Reformulación del Método de Programación Lineal Entera.

Entonces, teniendo en cuenta todos los resultados, quedan claras las ventajas de la Reformulación del Método de Programación Lineal Entera para resolver el Problema de Agregación de Rankings con menos recursos computacionales, o bien, resolver instancias que anteriormente no era posible resolver.

Referencias

Aledo, Juan A ; Gámez, José A ; Rosete, Alejandro: Approaching Rank Aggregation Problems By Using Evolution Strategies: The Case Of The Optimal Bucket Order Problem. European Journal Of Operational Research, 2018, Vol. 270, No. 3, P. 982–998

Aledo, Juan A ; Gámez, José A ; Rosete, Alejandro: A Highly Scalable Algorithm For Weak Rankings Aggregation. Information Sciences, 2021, Vol. 570, P. 144–171

Ali, Alnur ; Meila, Marina: Experiments With Kemeny Ranking: What Works When? Mathematical Social Sciences, 2012, Vol. 64, No. 1, P. 28–40

Chatterjee, Sujoy ; Mukhopadhyay, Anirban ; Bhattacharyya, Malay: A Weighted Rank Aggregation Approach Towards Crowd Opinion Analysis. Knowledge-Based Systems, 2018, Vol. 149, P. 47–60

Galdi, Paola ; Fratello, Michele ; Trojsi, Francesca ; Russo, Antonio ; Tedeschi, Gioacchino ; Tagliaferri, Roberto ; Esposito, Fabrizio: Stochastic Rank Aggregation For The Identification Of Functional Neuromarkers. Neuroinformatics, 2019, Vol. 17, No. 4, P. 479–496

Kaur, Parneet ; Wang, Gai-Ge ; Singh, Manpreet ; Singh, Sukhwinder: Rank Aggregation Using Moth Search For Web. En: International Conference On Innovative Computing And Communications : Springer, 2021, P. 63–74

Konobey: Linear Program Solver, [Consultado El: 17 De Agosto Del 2020]. Disponible En: Https://Sourceforge.Net/Projects/Lipside/

Lestari, Sri ; Adji, Teguh Bharata ; Permanasari, Adhistya Erna: Performance Comparison Of Rank Aggregation Using Borda And Copeland In Recommender System. En: 2018 International Workshop On Big Data And Information Security (Iwbis) : Ieee, 2018, P. 69–74

Liu, Yang ; Chen, Ting-Yu ; Yang, Zhi-Yan ; Fang, Wei ; Zhang, Chao: Identification Of Hub Genes In Thyroid Cancer: Robust Rank Aggregation And Weighted Gene Co-Expression Network Analysis. Disponible En: Ssrn 3502353, 2019

Mattei, Nicholas ; Walsh, Toby: Preflib: A Library For Preferences Http://Www.Preflib.Org. Perny, P. ; Pirlot, M. ; Tsoukiàs, A. (Hrsg.): Algorithmic Decision Theory - Third International Conference, Adt 2013, Bruxelles, Belgium, November 12-14, 2013, Proceedings, Lecture Notes In Computer Science, 2013, Vol. 8176, P. 259–270

Onan, Aytuğ ; Korukoğlu, Serdar: A Feature Selection Model Based On Genetic Rank Aggregation For Text Sentiment Classification. Journal Of Information Science, 2017, Vol. 43, No. 1, P. 25–38

Rosete-Suárez, Alejandro: Reformulación Eficiente Del Problema De Programación Lineal De Agregación De Rankings. Ingeniería Industrial, 2018, Vol. 39, P. 250–260

Subochev, Andrey ; Aleskerov, Fuad ; Pislyakov, Vladimir: Ranking Journals Using Social Choice Theory Methods: A Novel Approach In Bibliometrics. Journal Of Informetrics, 2018, Vol. 12, No. 2, P. 416–429

Tao, Zhifu ; Liu, Xi ; Zhou, Ligang ; Chen, Huayou: Rank Aggregation Based Multi-Attribute Decision Making With Hybrid Z-Information And Its Application. Journal Of Intelligent & Fuzzy Systems, 2019 Vol. 37, No. 3, P. 4231–4239

Universidad De Constanza: Knime. [Consultado El: 02 De Mayo Del 2020]. Disponible En: Https://Www.Knime.Com/Downloads

Xiao, Yu ; Deng, Ye ; Wu, Jun ; Deng, Hong-Zhong ; Lu, Xin: Comparison Of Rank Aggregation Methods Based On Inherent Ability. In: Naval Research Logistics (Nrl), 2017, Vol. 64, No. 7, P. 556–565

Z. I. Berlin: Scip: Solving Constraint Integer Programs. [Consultado El: 22 De Abril Del 2020]. Disponible En: Https://Www.Scipopt.Org/

Conflicto de interés

Los autores no reconocen tener ningún conflicto de intereses respecto al trabajo.

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 1, Enero-Marzo, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 62-76

Contribuciones de los autores

1. Conceptualización: Randy Reyna Hernández, Alejandro Rosete Suárez
2. Curación de datos: Randy Reyna Hernández
3. Análisis formal: Randy Reyna Hernández, Alejandro Rosete Suárez
4. Investigación: Randy Reyna Hernández, Alejandro Rosete Suárez
5. Metodología: Randy Reyna Hernández, Alejandro Rosete Suárez
6. Administración del proyecto: Randy Reyna Hernández, Alejandro Rosete Suárez
7. Recursos: Randy Reyna Hernández, Alejandro Rosete Suárez
8. Software: Randy Reyna Hernández
9. Supervisión: Alejandro Rosete Suárez
10. Validación: Randy Reyna Hernández
11. Visualización: Randy Reyna Hernández
12. Redacción – borrador original: Randy Reyna Hernández
13. Redacción – revisión y edición: Randy Reyna Hernández, Alejandro Rosete Suárez