

An ad-hoc algorithm to find the minimum winning coalition

Ignacio Lincolao-Venegas*, Eduardo Lobos-Pacheco†, Pedro Mirabal*, Antonio Parra-Riquelme†,
Victor Quiroz-Valenzuela †, Julio Rojas-Mora*

*Departamento de Ingeniería Informática

Universidad Católica de Temuco

Email: [ilicolao,pedro.sanchez,julio.rojas]@uct.cl

†Carrera de Ingeniería Civil en Informática

Universidad Católica de Temuco

Email: [elobos2018,aparra2018,vquiroz2018]@alu.uct.cl

Abstract—Finding the minimum winning coalition (MWC) is a particular case of clustering problems constrained on the clusters' size. As such, this problem is NP-Hard, posing an exciting challenge in optimization. In this work, we present a new *ad-hoc* algorithm to solve the MWC problem, which identifies the same MWC found by a genetic algorithm implemented under the same platform. The algorithm works deterministically and achieves a solution in tenths of a second for a real problem with high complexity, the political spectrum of the House of the 75° Congress of the USA as calculated with DW-Nominate. Hence, the solution time is two orders of magnitude better than the average time it takes for the genetic algorithm to find it.

Index Terms—Minimum winning coalition, constrained clustering, constrained optimization, binary optimization.

I. INTRODUCCIÓN

Según Riker [1, p. 12], se define una coalición como un subgrupo, con reglas aceptadas por todos sus miembros, que puede decidir por el grupo completo. Kelley [2] establece como condiciones para dicha coalición el que deben perseguir un objetivo común y articulado, agrupar sus recursos relevantes, participar en comunicación consciente relativa al objetivo, y acordar la distribución de la recompensa recibida cuando se alcance el objetivo.

Riker [1, p. 33] argumenta que en un juego n -personal de suma cero, con pagos complementarios y con jugadores racionales que poseen información perfecta, las coaliciones formadas sólo pueden ser coaliciones ganadoras mínimas (CGM). Esto implica que las coaliciones formadas en entornos de decisión deben tener el tamaño más pequeño posible para asegurar la victoria. La solución de este problema, es decir -la identificación de la CGM- se ha enfocado, tradicionalmente, desde la perspectiva de la teoría de juegos [3], [4]. Para un grupo de personas, la utilidad obtenida por formar parte de la coalición

ganadora no es sencilla de calcular, pues lo que se observa es la manifestación de una variable latente o de un constructo; la decisión. Por tanto, identificar la CGM utilizando herramientas derivadas de la teoría de juegos se convierte en un problema complejo. Una alternativa se centra en la metodología desarrollada por Poole y Rosenthal [5] para situaciones en las que los legisladores¹ estudiados tienen que seleccionar una de dos posiciones sobre un tema propuesto. Se asume que un legislador tiene un punto ideal en un espacio político latente y votará por la alternativa más cercana al mismo, sujeto a un error aleatorio. Por su imposibilidad práctica, en vez de definir una función de utilidad del legislador se estima el punto ideal a partir de las votaciones en las que el legislador toma parte. El algoritmo ubica al legislador en un espacio bidimensional del que se puede calcular la *distancia política*, un concepto que DeSwaan [6] ya utilizaba en el contexto de teoría de juegos. A partir de estas posiciones y midiendo la distancia euclídea política, se puede calcular la coalición ganadora mínima de una forma similar al Min Sum 2-Clustering Euclídeo Cuadrático [7]. No obstante, [7]–[9] exponen que este es un problema NP-Hard, por lo que sólo utilizando una heurística adecuada es posible resolverlo de manera eficiente. El objetivo de este trabajo es presentar una heurística *ad-hoc* la cual permite determinar la CGM de forma eficiente, es decir, en el menor tiempo posible dada la complejidad del problema.

II. ESTADO DEL ARTE

Las técnicas de agrupamiento (*clustering*) nos permiten resolver múltiples tipos de problemas, generando una gran cantidad de aplicaciones y variaciones de esta misma. Estos problemas pueden ser de complejidad NP-completo o de NP-Hard, incluso si estos son con restricciones dentro de un espacio euclidiano. Es por eso que a lo largo del tiempo para cada problema de agrupamiento se ha estudiado su nivel de complejidad. En [8] se estudia la complejidad de múltiples problemas de agrupamiento, describiendo aquellos que son

El equipamiento usado en este trabajo ha sido financiado con el proyecto HPC-Cluster UCT: Una iniciativa Interfacultades para el fortalecimiento de la investigación, vinculación con el medio, y creación de redes de forma interdisciplinaria (VIP FEQUIP2019-INRN-03).

¹Los términos legislador, congresista, representante y parlamentario son usados indistintamente en este trabajo.

NP-completos. En [9] se realizan estudios de agrupamiento de conjuntos, donde resolver un problema de $\sum \sum ||x - m||^2$ lo declaran como NP-Hard. Posteriormente, en [7] se analiza la complejidad de problemas de partición de un conjunto finito de puntos del espacio euclidiano en dos grupos, identificando como NP-Hard al problema Min-Sum 2-clustering. En [10] se profundiza esta línea de estudio, identificando casos denominados como fuertemente NP-Hard.

Debido a la complejidad que conlleva resolver los problemas de agrupamiento, se han utilizados distintos algoritmos heurísticos para resolver estos problemas. Por ejemplo, en [11] se usa el algoritmo de búsqueda tabú para agrupar m puntos en c grupos tal que la distancia entre los puntos dentro de un grupo y su centroide sea minimizada. En [12] utilizan una mezcla entre recocido simulado y el algoritmo K-means armónico para evitar mínimos locales en la solución de problemas de agrupación. Otros trabajos, como [13]–[15], utilizan algoritmos genéticos. También existen algoritmos basados en enjambres como el “firefly algorithm” utilizado en [16] realizando agrupamiento particional. Su rendimiento en comparación con otros algoritmos de enjambre fue superior al generar centros de conglomerados óptimos. El algoritmo “Artificial Bee Colony” se utilizó en [17] con este mismo fin, mientras que el “Honey-bee Mating optimization algorithm” fue utilizado en [18].

Existen otros algoritmos, como los basados en evolución universal [19], que utilizan analogías del “big bang” y el “Big Crunch”. Los resultados de este trabajo demostraron que ésta es una técnica adecuada para la agrupación de datos, además de tener una estructura simple y proporcionar clústeres de alta calidad en términos de suma de distancias dentro del cluster. Por otra parte, el algoritmo “Black Hole” [20] es un algoritmo que combina al algoritmo búsqueda gravitacional y el K-means. Se reportan en este trabajo resultados alentadores en cuanto a la calidad de las soluciones y la velocidad de convergencia del algoritmo propuesto.

III. DEFINICIÓN DEL PROBLEMA

Nuestro problema es una variación de [7], [10], es decir, la identificación de agrupamientos sujetos a restricciones de tamaño. No obstante, no centramos nuestro análisis únicamente en el centroide, sino que, como veremos más adelante, dividimos el proceso de identificación de la agrupación de interés en un algoritmo no basado en el centroide y un algoritmo de refinamiento por búsqueda local basado en el centroide.

Dado el conjunto de los n congresistas que conforman el parlamento

$$P = \{p_1, p_2, p_3, \dots, p_n\} \quad (1)$$

donde $p_i = \langle X_i, Y_i \rangle$ la puntuación de las dos primeras dimensiones definidas por DW-Nominate para cada miembro del parlamento en una votación dada. Se define $d(p_i, p_j)$ como la distancia euclidiana entre p_i y p_j .

Sea además q el quórum necesario para formar una posible coalición ganadora, por lo que la cantidad de coaliciones posibles se puede calcular a partir como un coeficiente binomial:

$$C(n, q) = \binom{n}{q} = \frac{n!}{q! \cdot (n - q)!} \quad (2)$$

Para conseguir la CGM del congreso P con un quorum q , es necesario encontrar a G^* , subconjunto de P :

$$G = \{g_1, g_2, \dots, g_q\} \subset P, \quad (3)$$

cuyo tamaño es q , para el cual la sumatoria de la distancia entre todos sus miembros sea mínima:

$$Z(G) = \sum_{i=1}^{q-1} \sum_{j=i+1}^q d(g_i, g_j) \quad (4)$$

$$G^* = \arg_G \min Z(G) \quad (5)$$

Podemos observar que este problema es similar al Min Sum 2-Clustering Euclídeo Cuadrático de [7], en el que nos interesa encontrar solo una de las agrupaciones: G^* . Aunque como veremos más adelante el polígono convexo de ambas agrupaciones pueda parecer solapado o cortado por el otro, estas son disjuntas, en el sentido de que cada elemento en P pertenece o no pertenece a G^* .

IV. SOLUCIÓN DEL PROBLEMA

Dado que el modelo anteriormente descrito es NP-Hard, presentamos un algoritmo *greedy* que permite identificar, de manera determinista, una coalición ganadora (CG) que converge a la encontrada por metaheurísticas como los algoritmos genéticos en una fracción del tiempo y que consideramos como una buena aproximación a la CGM.

Usando P , encontramos G_i , que es una CG en la que sus integrantes suman la menor distancia respecto a p_i :

$$G_i = \{g_i, g_{[1]}, \dots, g_{[q-1]}\} : \forall i, j = 1, \dots, n, \quad (6)$$

donde $g_{[j]}$ es el j -ésimo elemento más cercano a g_i que no incluye a g_i . Por lo tanto:

$$|G_i| = q. \quad (7)$$

Para todos los p_i se calcula su respectivo G_i y se construye el conjunto de tuplas:

$$\mathbb{G} = \{\langle G_i, Z(G_i) \rangle : i = 1, \dots, n\}, \quad (8)$$

donde $Z(G_i)$ es el valor de la función objetivo (Eq. 4) para G_i (*fitness* de G_i , a partir de ahora). Usando $Z(G_i)$ identificamos como solución inicial o candidata a:

$$G^c = \arg_{G_i} \min Z(G_i), \quad (9)$$

es decir, la CG con el menor *fitness* de aquellas en \mathbb{G} . La complejidad de este algoritmo es $O(n^2 \log(n))$, pues se realizan n ordenamientos de complejidad $O(n \log(n))$.

A continuación se buscan soluciones mediante un algoritmo de búsqueda local (ver Algoritmo 1 que sustituye congresistas del polígono convexo H que rodea a G^c por los congresistas P^c que se encuentren fuera de este, como máximo a una distancia αr del centroide C de G^c , donde $\alpha > 1$ y r es la distancia máxima de C a los puntos en H , y que han sido ordenados de manera decreciente de acuerdo a esta distancia.

Algorithm 1 Algoritmo de búsqueda local

Entrada: P, G^c, α

Salida: G^*

```

1:  $G^* \leftarrow G^c$ 
2:  $C \leftarrow \text{Centroide de } G^c$ 
3: repeat
4:    $G_{orig}^* \leftarrow G^*$ 
5:    $H \leftarrow \text{Polígono convexo de } G^c$ 
6:    $R \leftarrow \{d(h'_j, C) : \forall h'_j \in H'\}$ 
7:    $H' \leftarrow H$  ordenado decrecientemente
    según su distancia a  $C$ 
8:    $r \leftarrow \alpha \max R$ 
9:    $P' \leftarrow P \setminus G^*$ 
10:   $P^c \leftarrow \text{Conjunto ordenado crecientemente de los } P \text{ que}$ 
    no pertenecen a  $G^*$  y que se encuentran dentro de  $r$ 
11:   $j \leftarrow 1$ 
12:  while  $j \leq |H'|$  or  $Z(G^c) < Z(G^*)$  do
13:    for  $p_k^c \in P^c$  do
14:      Cambio  $h'_j$  por  $p_k^c$  en  $G^c$ 
15:      Calcular  $Z(G_k^c)$ 
16:    end for
17:     $G^c \leftarrow \arg \min_{G_k^c} Z(G_k^c)$ 
18:     $j = j + 1$ 
19:  end while
20:  if  $Z(G^c) < Z(G^*)$  then
21:     $G^* \leftarrow G^c$ 
22:    Actualizar  $C$ 
23:  end if
24: until  $G^* == G_{orig}^*$ 

```

A partir de H se construye H' , el conjunto de congresistas que forman parte del polígono convexo de G^c ordenados de manera decreciente de acuerdo a su distancia al centroide C de G^c . Se selecciona un congresista h'_j de H' , quien es sustituido por el congresista p_k^c de P^c para formar una nueva CG, G_k^c . De entre las k CG obtenidas por este proceso, G_k^c , se selecciona aquella que tiene el mejor *fitness* y si esta es más baja que la de la coalición candidata G^* se reemplaza esta última por la primera. Si esto sucede el ciclo se detiene y se vuelve al inicio del algoritmo para recalcular C , H' , r y P^c , procediendo a comparar nuevamente los congresistas de H' con los de P^c . Si no se consigue ninguna solución candidata que mejore la que actualmente se considera como la CGM G^* , se procede a realizar el mismo proceso con el siguiente congresista en H' . Si al recorrer H' no se actualiza G^* , el algoritmo se detiene.

La complejidad computacional en tiempo de cada iteración

del Algoritmo 1 está acotada por $O(|H|^2)$ como peor caso, siendo $|H|$ la cardinalidad del polígono convexo G^c . El escenario de estar ante el peor caso es altamente improbable, pues implicaría que todos los puntos de G^c estarían sobre el polígono convexo a la misma distancia del centroide. Si detallamos los pasos más relevantes del Algoritmo 1, tenemos en la línea 2 el cálculo del centroide de G^c . Este puede ser computado en tiempo $O(|G^c|)$ y luego actualizado en tiempo constante en cada iteración. En la línea 5 se calcula el Polígono convexo H utilizando el método de Graham [21] en tiempo $O(|G^c| \log |G^c|)$, siendo este el término dominante en cuanto a complejidad temporal del Algoritmo 1 en su caso promedio.

V. CASO DE ESTUDIO

Como caso de estudio para probar el algoritmo se utilizaron datos de la Cámara de Representantes del Congreso de los Estados Unidos de América. Dos razones nos llevaron a ello. En primer lugar, esta instancia parlamentaria está constituida por 435 representantes [22], lo que haría imposible resolver el problema por algoritmos deterministas tradicionales como branch-and-bound, incluso usando software del estado del arte, como CPLEX, Gurobi o SCIP [23]. Debemos recordar que en el peor de los casos, branch-and-bound tiene una complejidad exponencial y no se diferencia de un algoritmo de fuerza bruta. En el caso promedio es también exponencial, aunque con una complejidad mucho menor [24]. Por lo tanto, este cuerpo legislativo se convertía en un campo de pruebas ideal para nuestro algoritmo.

En segundo lugar, necesitábamos datos del espectro político de un cuerpo legislativo. Esta información se encuentra disponible a través de Voteview.com [25]. Esta iniciativa entrega las dos primeras dimensiones del espectro político calculadas por el algoritmo DW-Nominate [26] para todas las votaciones de la Cámara de Representantes del Congreso de los Estados Unidos de América. Por lo tanto, no era necesario realizar estos cálculos nosotros mismos, lo que nos permitía centrarnos en la programación y prueba de nuestro algoritmo.

Debido a que de todos los datos disponibles necesitábamos un caso de la mayor complejidad posible, se debía seleccionar un caso de estudio con posiciones políticas poco polarizadas. La razón radica en que mientras más cercanos se encuentren los partidos que forman un parlamento, las zonas de solapamiento permitirán evaluar coaliciones que incluyan diputados de todos ellos. Aunque el sistema político estadounidense ha sido tradicionalmente bipartidista, se pueden observar diferentes grados de polarización a lo largo de la historia. De acuerdo a [27], [28] el 75° Congreso de los Estados Unidos de América, que sesionó entre el 3 de enero de 1975 y el 3 de enero de 1977, conformaría el mejor caso de estudio, dado que fue el menos polarizado desde 1970. La Cámara estaba conformada por un 66 % de congresistas del Partido Demócrata y un 34 % del Partido Republicano, por lo que con tal supermayoría, una coalición bipartidista sería básicamente imposible de observar, dada la gran legitimidad que tendrían las decisiones que los demócratas tomaran.

Por inspección visual seleccionamos de todas las votaciones la

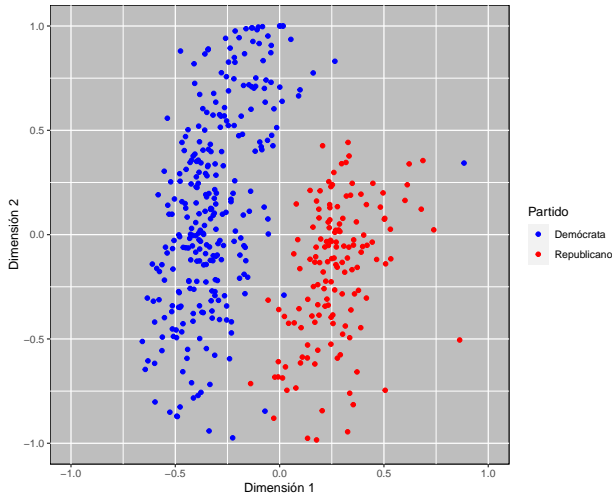


Figura 1. Votación RH0941234

RH0941234 (Proyecto de Ley Nro. HRE1553) de la Cámara. En esta votación tomaron parte 431 representantes (de un total de 435), por lo que la mayoría absoluta, q , estaría compuesta por 216 representantes. Como puede apreciarse en la Figura 1, la varianza en la primera dimensión es baja en el Partido Demócrata, con la sola excepción del congresista Lawrence Patton McDonald, representante del estado de Georgia y uno de los miembros del grupo llamado los “demócratas del sur” o “dixiecrats”, la extrema derecha de este partido que a partir de los años 60 del siglo pasado comenzó a fusionarse con el Partido Republicano. En la segunda dimensión del espectro político, por el contrario, existe una gran dispersión entre los miembros del Partido Demócrata, lo que indica que en ciertas votaciones hubo desacuerdos que los separaban. En este sentido cabe hacerse la pregunta de si una CGM bipartidista centrista tendría una estabilidad mayor que un acuerdo unipartidista, sobre todo en temas que tocasen la segunda dimensión². A continuación, veremos cómo responde a esta pregunta la definición de la CGM que hacemos mediante la función objetivo definida en (4) bajo la restricción $|G^*| = q$.

VI. IMPLEMENTACIÓN

El algoritmo fue implementado en C++³ utilizando Cmake 3.22.1 y GCC 7.5.0 bajo el sistema operativo openSUSE Leap 15.2. La compilación y ejecución se realizaron en un servidor Lenovo ThinkSystem SR530 con dos procesadores Intel Xeon Gold 5220 de 18 núcleos cada uno a una velocidad base de 2,20 GHz (ver Cuadro I). Usando esta misma configuración de

²DW-Nominate no hace valoraciones sobre lo que representan las dimensiones del espectro político, únicamente identifica dimensiones de separación entre los parlamentarios. Sin embargo, en Voteview.com etiquetan la primera dimensión como relacionada a temas económico-redistributivos, mientras que la segunda dimensión acumularía todos los demás temas. Estas etiquetas son colocadas *ex-post*, luego de un análisis de los temas que alejan o acercan a los parlamentarios.

³Una versión funcional del algoritmo puede ser descargada de <https://github.com/vicquiroz/MWC.git>.

Cuadro I
ESPECIFICACIONES DE HARDWARE Y SOFTWARE

Hardware	Especificación
Memoria RAM	
Capacidad	16 Gb x4 (64 Gb)
Velocidad base(MHz)	2933
Tipo de Memoria	DDR4
CPU	
Modelo de CPU	2 Intel Xeon Gold 5220
Cantidad de núcleos	18
Cantidad de hilos	36
Base Clock(GHz)	2,20
Software	Especificación
Sistema Operativo	openSUSE Leap 15.2
Cmake	3.22.1
GCC	7.5.0

software y hardware se realizó una implementación de un algoritmo genético, a manera de *baseline* para las comparaciones posteriores.

VII. RESULTADOS

Como hemos explicado anteriormente, dado el número de representantes que participaron activamente de la votación estudiada, es decir 431 miembros, la CGM debe tener una cardinalidad $q = 216$ miembros para una mayoría absoluta.

En la Figura 2 se puede observar la frecuencia con que cada i aparece en la totalidad de las 431 CG que se construyen inicialmente. Las zona amarilla muestra parlamentarios que aparecen más frecuentemente que aquellos de las zonas violeta. Se advierte que dada la naturaleza geométrica del problema y el parcial solapamiento de los representantes de los dos partidos políticos, los que se encuentran en zonas centrales, principalmente del Partido Demócrata, se constituyen en buenos candidatos a pertenecer a la CGM.

Si observamos ahora el *fitness* de cada CG, apreciamos que aquellas centradas en los representantes centrales del Partido Demócrata son mejores (tonos violeta, en la centro-izquierda) que los restantes (ver Figura 2).

Finalmente, para observar simultáneamente el *fitness* de cada G_i^c con la frecuencia de aparición en las G_i^c de cada representante i (puntos blancos), usamos un gráfico de acuarela [29] (ver Figura 4). Este gráfico fue construido a partir de 10000 regresiones tipo kernel seleccionando mediante *bootstrap* las submuestras de los 431 representantes que se usarían para cada una de ellas. Las zonas más blancas indican una mayor concentración de regresiones que pasan por ellas que las más

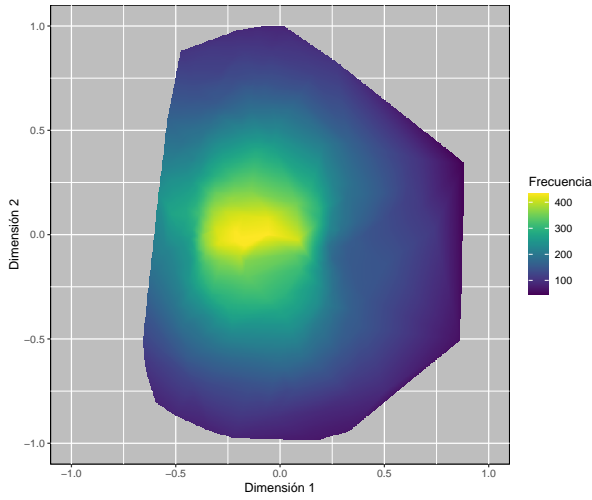


Figura 2. Frecuencia de cada i en las G_i^c .

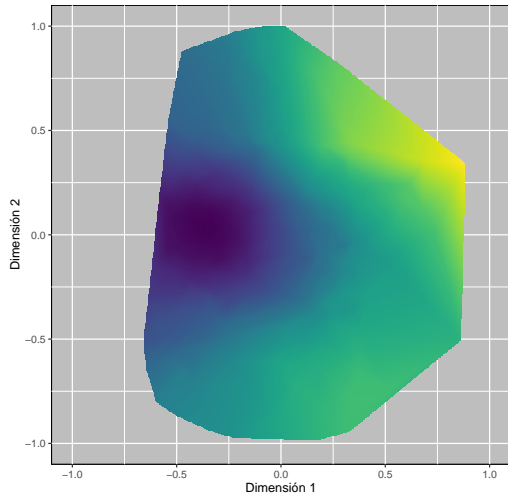


Figura 3. $Fitness$ de las G_i^c para cada i .

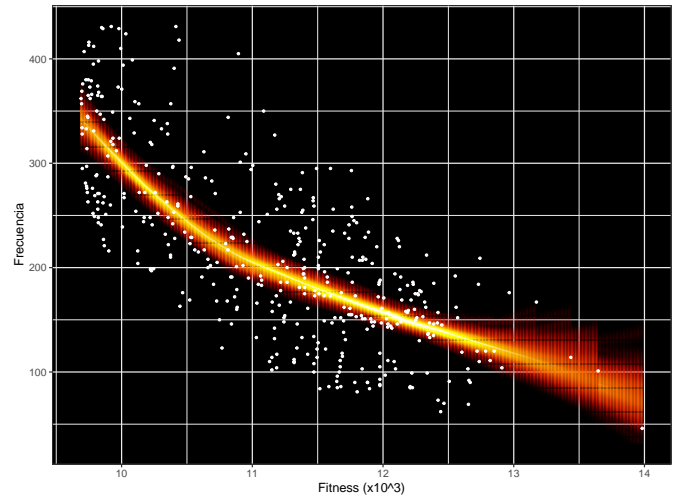


Figura 4. Regresión tipo kernel con bootstrap entre las $fitness$ de las G_i^c para cada i y la frecuencia de aparición de cada i en las G_i^c .

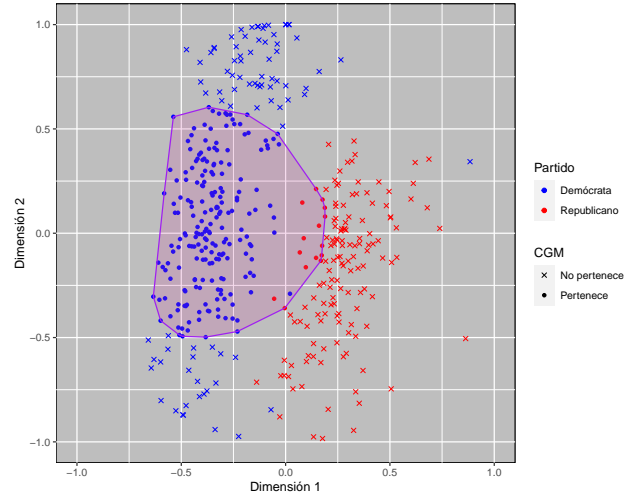


Figura 5. Solución inicial G^c y polígono convexo H

naranjas o rojas. Se aprecia una relación decreciente entre ambos indicadores. Las CG con mejor $fitness$ suelen estar unidas a parlamentarios con una mayor probabilidad de aparición en las CG formadas a partir de los demás representantes.

El algoritmo de identificación de la solución candidata inicial seleccionó la presentada en la Figura 5, aquella correspondiente al representante demócrata por el estado de Illinois, Tim Lee Hall. Como se observa, la G^c está compuesta básicamente por miembros del Partido Demócrata y solo 15 miembros del Partido Republicano, con $Z(G^c) = 9689,948859$.

El algoritmo genético implementado como *baseline* y configurado de acuerdo a los parámetros presentados en el Cuadro II, obtuvo, a partir de poblaciones iniciales aleatorias, una CGM con $Z(G^*) = 9686,93831$. Esta CGM fue alcanzada un 91,41 % de las 10000 iteraciones realizadas con una desviación estándar de 0,9017. Esto implica que la solución inicial de

nuestro algoritmo es muy cercana a esta solución y, como veremos a continuación, converge a ella en la segunda parte del algoritmo.

En la Figura 6, correspondiente a la primera iteración del algoritmo de búsqueda local, se localiza el centroide del polígono convexo de la G^c (punto amarillo). El representante que forma parte del polígono convexo de G^c y que está más alejado del centroide es Gillis William Long, Demócrata por el estado de Louisiana. Con un $r = 0,62660$ se seleccionan 47

Cuadro II
PARÁMETROS

Tamaño de Población	Probabilidad de Mutación	Probabilidad de Selección
38	0,1700019	0,141

Cuadro III
RESULTADOS

Resultado esperado	Precisión	Desviación estándar
9686,93831	91,41 %	0,9017
Promedio Iteraciones	Desviación estándar Iteraciones	
15315,53	2326,414	
Promedio Tiempo (s)	Desviación estándar Tiempo	
26,50445	4,082715	

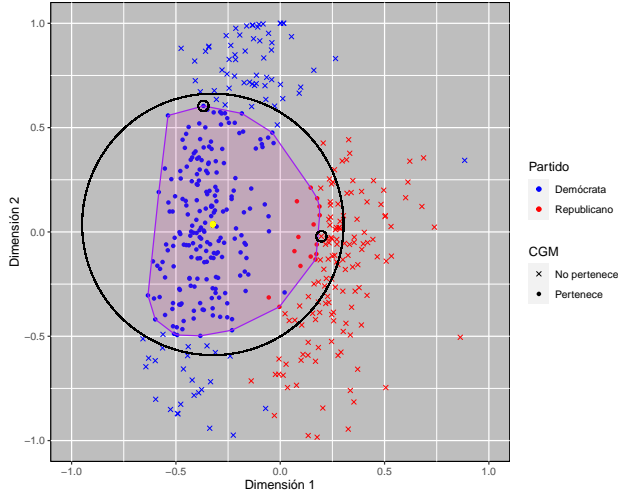


Figura 6. Solución G^c y polígono convexo H , primera iteración del algoritmo de búsqueda local

representantes en P^c , es decir, aquellos que no forman parte de G^c y se encuentran dentro de este radio. De estos se identificó al representante Albert W. Johnson, Republicano por el estado de Pennsylvania como el sustituto de Long, mejorando el fitness a $Z(C^g) = 9688,493662454$. El cambio de un Demócrata del Sur por un Republicano del Este es notable, dadas las diferencias políticas que a nivel geográfico que caracterizan al Congreso de los Estados Unidos de América.

La segunda iteración, correspondiente a la Figura 7, determina que el miembro del polígono convexo de G^c que debe salir la CG es Henry B. González, Demócrata por el estado de Texas. Usando $r = 0,562433823$ se seleccionaron 9 representantes como candidatos a entrar en la CG. El seleccionado fue el Republicano por New Jersey, Matthew John Rinaldo, llevando el valor del fitness a $Z(C^g) = 9686,938309982$, el mismo valor alcanzado con el algoritmo genético, lo que indica una convergencia a la misma solución. Nuevamente, se observa la sustitución de un Demócrata del Sur por un Republicano del Este, lo que apunta a la ya citada diferencia política que a nivel geográfico se observa en los Estados Unidos de América.

En la última iteración (ver Figura 8), el algoritmo determina que $r = 0,497197910$. Dentro del mismo no se encuentra ningún representante que pueda ser candidato a entrar en C^g , por lo que entendemos haber llegado a la CGM (ver Figura 9). Por lo tanto, de los 216 representantes en ella, se pasó de 15 a 17 representantes del Partido Republicano.

El algoritmo en ambas etapas ha logrado converger a la mis-

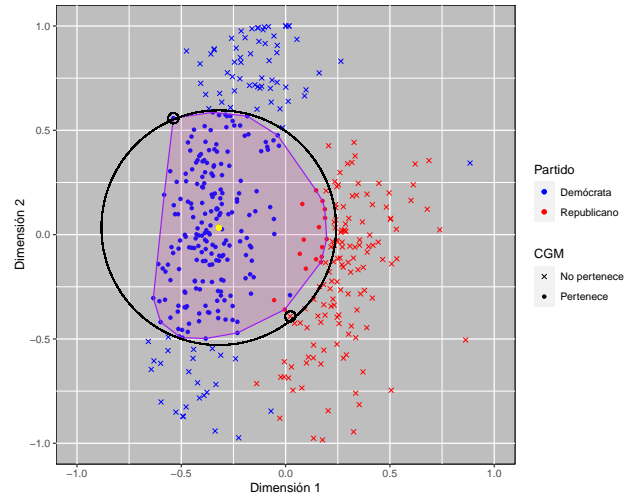


Figura 7. Solución G^c y polígono convexo H , segunda iteración del algoritmo de búsqueda local

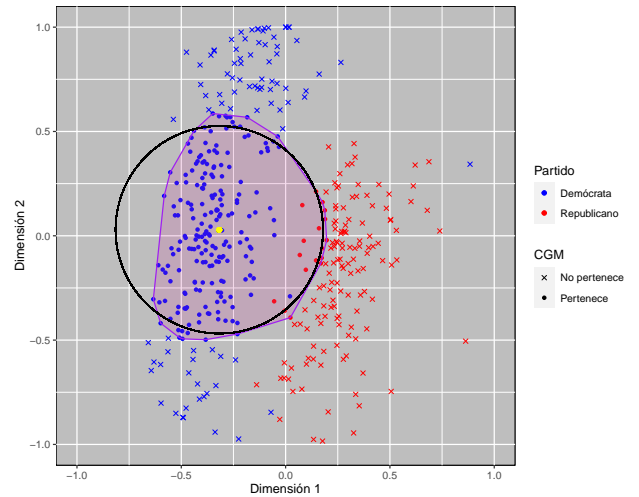


Figura 8. Solución G^c y polígono convexo H , tercera iteración del algoritmo de búsqueda local

ma solución que el algoritmo genético usado como *baseline* en 0,1s, mientras que este último, como se observa en el Cuadro III, lo ha hecho en un tiempo promedio de 26,50445s con una desviación estándar de 4,082715s, es decir, una mejora de tres órdenes de magnitud en el tiempo de identificación de la CGM.

VIII. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo presentamos un nuevo algoritmo *ad-hoc* para solucionar el problema de la CGM. Este problema, un caso particular de problema de agrupamiento con restricciones en el tamaño de las agrupaciones, es de gran interés para el análisis del funcionamiento de cuerpos deliberativos. Nuestro algoritmo, identifica la misma CGM que la encontrada por una implementación bajo la misma plataforma de un algoritmo

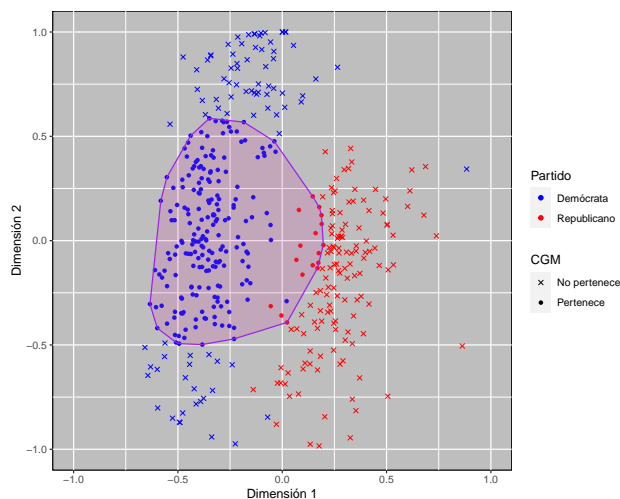


Figura 9. Solución final G^* y polígono convexo H

genético. El algoritmo funciona de manera determinista y consigue esta solución en décimas de segundo, para un problema real con una complejidad elevada. El tiempo de solución es de dos órdenes de magnitud mejor que el tiempo promedio que le toma al algoritmo genético encontrarla.

Como trabajo futuro nos planteamos dos vías. La primera vía nos llevaría a probar con casos más pequeños y artificialmente contruidos, que puedan ser resueltos mediante branch-and-bound o fuerza bruta, para corroborar que la convergencia a la CGM existe, incluso en casos complejos. Con ese trabajo también buscaríamos determinar la sensibilidad de la solución a cambios tanto de los puntos del polígono convexo que delimita la CGM como de puntos fuera de la CGM. Por medio de la segunda vía analizaríamos la Cámara de Diputadas y Diputados de la República de Chile, para identificar patrones de disciplina partidista y asociaciones entre diputados y diputadas de gobierno y oposición en las decisiones tomadas, especialmente en votaciones de gran connotación social. [30]

REFERENCIAS

- [1] W. H. Riker, *The Theory of Political Coalitions*. Yale University Press, 1962.
- [2] E. W. Kelley, "Techniques of studying coalition formation," *Midwest Journal of Political Science*, pp. 62–84, 1968.
- [3] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*, 2nd rev. Princeton University Press, 1947.
- [4] A. V. Kel'manov and A. V. Pyatkin, "NP-Hardness of some Quadratic Euclidean 2-clustering problems," *Doklady Mathematics*, vol. 92, no. 2, pp. 634–637, 2015.
- [5] P. Brucker, "On the complexity of clustering problems," in *Optimization and operations research*. Springer, 1978, pp. 45–54.
- [6] T. F. Gonzalez, "On the computational complexity of clustering and related problems," in *System modeling and optimization*. Springer, 1982, pp. 174–182.

- [4] M. A. Leiserson, *Coalitions in politics: A theoretical and empirical study*. Yale University, 1966.
- [5] K. T. Poole and H. Rosenthal, "A spatial model for legislative roll call analysis," *American Journal of Political Science*, pp. 357–384, 1985.
- [6] A. De Swaan, "An empirical model of coalition formation as an n-person game of policy distance minimization," *The study of coalition behavior*, pp. 424–444, 1970.
- [10] A. Kel'manov, A. Pyatkin, and V. Khandeev, "Np-hardness of quadratic euclidean 1-mean and 1-median 2-clustering problem with constraints on the cluster sizes," in *Doklady Mathematics*, vol. 100, no. 3. Springer, 2019, pp. 545–548.
- [11] K. S. Al-Sultan, "A tabu search approach to the clustering problem," *Pattern recognition*, vol. 28, no. 9, pp. 1443–1451, 1995.
- [12] Z. Güngör and A. Ünler, "K-harmonic means data clustering with simulated annealing heuristic," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 199–209, 2007.
- [13] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 825–832, 1996.
- [14] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.
- [15] P. Kudova, "Clustering genetic algorithm," in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*. IEEE, 2007, pp. 138–142.
- [16] J. Senthilnath, S. Omkar, and V. Mani, "Clustering using firefly algorithm: performance study," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 164–171, 2011.
- [17] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 652–657, 2011.
- [18] M. Fathian, B. Amiri, and A. Maroosi, "Application of honey-bee mating optimization algorithm on clustering," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1502–1513, 2007.
- [19] A. Hatamlou, S. Abdullah, and M. Hatamlou, "Data clustering using big bang–big crunch algorithm," in *International conference on innovative computing technology*. Springer, 2011, pp. 383–388.
- [20] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information sciences*, vol. 222, pp. 175–184, 2013.
- [21] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Info. Proc. Lett.*, vol. 1, pp. 132–133, 1972.
- [22] United States House of Representatives, "The House Explained." [Online]. Available: <https://www.house.gov/the-house-explained>
- [23] O. Şeker, N. Tanoumand, and M. Bodur, "Digital annealer for quadratic unconstrained binary optimization: a comparative performance analysis," *Applied Soft Computing*, vol. 127, p. 109367, 2022.
- [24] N. Thakoor, V. Devarajan, and J. Gao, "Computation complexity of branch-and-bound model selection," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1895–1900.
- [25] J. B. Lewis, "Voteview — Search." [Online]. Available: <https://voteview.com/>
- [26] K. Poole, *Spatial Models of Parliamentary Voting*. Cambridge University Press, 2005. [Online]. Available: <https://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521617475>
- [27] S. M. Theriault, "Party polarization in the us congress: Member replacement and member adaptation," *Party Politics*, vol. 12, no. 4, pp. 483–503, 2006.
- [28] Z. P. Neal, "A sign of the times? weak and strong polarization in the us congress, 1973–2016," *Social Networks*, vol. 60, pp. 103–112, 2020.
- [29] F. Schönbrodt, "Visually weighted/ Watercolor Plots, new variants: Please vote!" [Online]. Available: <http://www.nicebread.de/visually-weighted-watercolor-plots-new-variants-please-vote/>
- [30] D. A. Armstrong, R. Bakker, R. Carroll, C. Hare, K. T. Poole, and H. Rosenthal, *Analyzing spatial models of choice and judgment with R*. CRC Press, 2014.