

**Algoritmos multi-objetivo para la
detección de comunidades: un
estado del arte**

Darian Horacio Grass Boada, Airlé Pérez
Suárez, Rafael Bello Pérez y Alejandro Rosete
Suárez

RT_034

febrero 2016



REPORTE TÉCNICO
**Minería
de Datos**

**Algoritmos multi-objetivo para la
detección de comunidades: un
estado del arte**

Darian Horacio Grass Boada, Airlé Pérez
Suárez, Rafael Bello Pérez y Alejandro Rosete
Suárez

RT_034

febrero 2016



Tabla de contenido

1.	Introducción	2
2.	Marco teórico	6
2.1.	Definición del problema	6
2.2.	Definición del concepto de comunidad	7
2.3.	Funciones objetivos en la detección de comunidades	8
2.3.1.	Funciones basadas en la Modularidad	8
2.3.2.	Funciones basadas en la teoría de grafo	10
2.3.3.	Aspectos a destacar en las funciones objetivos	11
3.	Métodos de solución de problemas	12
3.1.	Representación de la solución	15
3.1.1.	Representación basada en etiqueta	15
3.1.2.	Representación basada en localización adyacente	16
3.2.	Algoritmos metaheurísticos	16
3.2.1.	Algoritmo genético basado en ordenamiento de no dominancia (NSGA-II, por sus siglas en inglés)	17
3.2.2.	Algoritmo evolutivo multi-objetivo basado en descomposición (MOEA/D, por sus siglas en inglés)	18
3.2.3.	Operadores heurísticos	19
3.3.	Obtención de la solución final	21
3.3.1.	Distancia mínima-máxima	22
3.3.2.	Estrategia basada en la identificación de los puntos <i>Knee</i>	23
3.3.3.	Estrategia basada en votación mayoritaria	23
4.	Trabajo relacionado	24
4.1.	Algoritmos de detección de comunidades en redes estáticas	25
4.1.1.	MEA-CDPs	25
4.1.2.	BOCD	26
4.1.3.	MOGA-Net	26
4.1.4.	MOCD	28
4.1.5.	MOEA/D-Net	29
4.1.6.	HSA-CLS	29
4.1.7.	EFA	30
4.1.8.	NSGA-Net	31
4.1.9.	GAsMCD	32
4.1.10.	MOGCM	32
4.1.11.	MOLS-Net	33
4.1.12.	IMOQPSO	34
4.1.13.	Limitaciones o deficiencias presentes en los métodos descritos	35
4.2.	Algoritmos de detección de comunidades de redes con signo	36
4.2.1.	MODPSO	36
4.2.2.	MEAs-SN	37
4.2.3.	MOEA-SN	38
4.2.4.	Limitaciones o deficiencias presentes en los métodos descritos	39
4.3.	Algoritmos de detección de comunidades de redes dinámicas	39
4.3.1.	DYN-LSNNIA	40

4.3.2.	DYN-MOGA	41
4.3.3.	DYN-DMLS	41
4.3.4.	Limitaciones o deficiencias presentes en los métodos descritos	42
4.4.	Algoritmos de detección de comunidades de redes multicaracterísticas	42
4.4.1.	MultiMOGA	42
4.4.2.	Limitaciones o deficiencias presentes en los métodos descritos	43
5.	Conclusiones	44
	Referencias bibliográficas	45

Lista de figuras

1.	Ejemplo de la representación basada en etiqueta: (a) topología del grafo, (b) representación de la solución, (c) estructura de comunidades [1].	15
2.	Ejemplo de la representación basada en localización adyacente: (a) topología del grafo, (b) representación de la solución, (c) estructura de comunidades [2].	16
3.	Algoritmos de detección de comunidades multi-objetivo: Taxonomía según el tipo de representación.	17
4.	Procedimiento NSGA-II [3].	18
5.	Algoritmos de detección de comunidades multi-objetivo: Taxonomía según la metaheurística utilizada como método de solución de problemas.	20
6.	Ejemplo de cruzamiento uniforme.	21
7.	Ejemplo de cruzamiento de dos puntos.	21
8.	Ejemplo del método de selección distancia Min-Max. La solución final es la que identifica 4 comunidades, donde k es el número de comunidades [4].	23
9.	Ejemplo de la identificación del punto <i>Knee</i> . S_1 y S_2 representan las soluciones de mejor valor en la función f_1 y f_2 respectivamente. L denota la línea extrema entre ambas soluciones [5].	24
10.	Algoritmos de detección de comunidades multi-objetivo: Taxonomía según el tipo de redes que se aborda.	25

Algoritmos multi-objetivo para la detección de comunidades: un estado del arte

Darian Horacio Grass Boada¹, Airel Pérez Suárez¹, Rafael Bello Pérez² y Alejandro Rosete Suárez³

¹ Equipo de Investigaciones de Minería de Datos, CENATAV - DATYS, La Habana, Cuba.
dgrass,asuares@cenatav.co.cu

² Centro de Estudios de Informática, Universidad Central Marta Abreu de Las Villas, Cuba.
rbellop@uclv.edu.cu

³ Centro de Estudios de Ingeniería y Sistemas, Facultad de Ingeniería Informática, Instituto Superior Politécnico José Antonio Echeverría, La Habana, Cuba.
rosete@ceis.cujae.edu.cu

RT_034, Serie Gris, CENATAV - DATYS

Aceptado: 22 de febrero de 2016

Resumen. Entre los principales problemas dentro del Análisis de Redes Sociales está la detección de comunidades. Las mismas, pueden estar descritas en entornos solapados o difusos, presentar ponderaciones positivas o negativas que describan las relaciones entre los elementos, además de sufrir cambios a partir de la adición y/o eliminación de nodos y aristas. Uno de los enfoques más utilizados en la detección de comunidades es el de optimización, donde se busca identificar comunidades que cumplan alguna propiedad específica, lo cual crea un sesgo en las estructuras identificadas. En los últimos años se han desarrollado un conjunto de métodos basados en un enfoque de optimización multi-objetivo, los cuales miden diferentes propiedades en las comunidades. En este trabajo se presenta un análisis crítico de los algoritmos reportados en la literatura para la detección de comunidades, bajo un enfoque de optimización multi-objetivo, resaltándose las principales insuficiencias, así como los posibles trabajos futuros a desarrollar.

Palabras clave: detección de comunidades, optimización multi-objetivo, algoritmos evolutivos multi-objetivo.

Abstract. Community Detection is one of the most important problems in Social Networks Analysis. Communities can be overlapped or fuzzy and they can contain positive or negative weights to describe the relationships between the elements. In addition, communities can change due to the addition and/or deletion of nodes and edges. Community detection can be viewed as an optimization problem in which a set of communities, meeting a specified property, is identified. However, since the above mentioned approach attempts to optimize only one objective function, the obtained solutions could have a bias towards a particular community structure property. In order to overcome this drawback, several multi-objective optimization algorithms have been proposed; these algorithms measure the fulfillment, by the communities, of different properties at the same time. In this paper, the multi-objective optimization algorithms reported in the literature for the detection of communities are analyzed. The limitations of the above mentioned methods are discussed and some promising and interesting research lines are presented.

Keywords: community detection, multi-objective optimization, multiobjective evolutionary algorithms.

1. Introducción

El incremento del volumen y variedad de información que se encuentra digitalizada en bases de datos y otras fuentes ha crecido considerablemente en las últimas décadas. Este incremento, cuantitativo y cualitativo, se hace notable en contextos que involucran individuos, colectivos y organizaciones, así como las relaciones sociales entre los mismos. De esta forma, con el auge del uso de la red de redes (WWW, por sus siglas en inglés) se observa un crecimiento significativo de investigaciones y aplicaciones orientadas a la extracción de conocimientos en dominios caracterizados por relaciones sociales, conocido como Análisis de Redes Sociales (SNA, por sus siglas en inglés) [6].

SNA tiene gran aplicabilidad en diferentes contextos, como son las redes de blogs, las redes de intercambio de correos electrónicos, las redes de colaboración, la identificación de organizaciones terroristas y el análisis de la opinión pública, entre otros. De manera general, estos sistemas complejos muestran una organización modular que refleja cierta similitud o afinidad entre los elementos del mismo módulo o grupo. La tarea de identificar grupos que presenten esta propiedad es conocida como: detección de comunidades o agrupamiento en grafo [7], siendo por su utilidad en los últimos años, uno de los principales problemas estudiados en el contexto del Análisis de Redes Sociales [1].

Esta organización modular o en comunidades responde en parte a las características comunes de los elementos de la red y/o las relaciones o enlaces presentes en estos. Desde la perspectiva de la teoría de grafo, la red es modelada mediante un grafo, donde los vértices (nodos) representan los objetos y las aristas (enlaces) la interacción entre ellos. De esta forma, la detección de comunidades permite descubrir clases de objetos y estudiar la organización modular del sistema, tanto en términos de su estructura topológica, así como en función de la información descriptiva de los objetos de la red y sus relaciones. En este sentido, el problema constituye un problema de clasificación no supervisada al igual que el agrupamiento, debido a que, en principio, no se cuenta con información de las comunidades o grupos subyacentes en la red. De hecho, el objetivo es precisamente identificar la estructuración subyacente a partir de ciertos criterios de agrupamientos, propiedades de los datos y formas de comparación de los mismos [8].

Cabe señalar que muchos de los métodos que abordan la clasificación no supervisada se apoyan fundamentalmente en la Teoría Clásica de Conjuntos. De manera general se determinan los grupos mediante dos enfoques procedimentales. El primer enfoque, denominado extensional, consiste en dar la lista de los objetos que conforman el conjunto. Por otro lado, el enfoque denominado intencional, consiste en dar la propiedad que caracteriza a los objetos de dicho conjunto. Los algoritmos clásicos de agrupamiento utilizan el primer enfoque para dar los objetos que forman cada grupo, mientras que los algoritmos de agrupamiento conceptual utilizan el segundo procedimiento [9]. Los métodos de detección de comunidades abordados en el presente trabajo siguen un enfoque procedimental extensional.

En ocasiones el número de grupos es también un problema a resolver, sobre todo en un contexto no supervisado como la detección de comunidades. Por tanto, según el conocimiento o no del número de comunidades o grupos a detectar, el problema se puede clasificar como clasificación no supervisada restringida o libre, respectivamente [8].

Si bien no existe un consenso respecto a una definición de comunidad, en principio se busca, intuitivamente, que el número de enlaces dentro de las comunidades sea superior que el número de enlaces presentes entre las mismas [10]. La no existencia de un consenso se debe en parte a: (1) la naturaleza no supervisada del problema a abordar, donde no se cuenta normalmente con

información previa que describa las comunidades idóneas subyacentes en la red, (2) cada dominio de aplicación puede presentar sus requerimientos y/o comprensiones acerca de las comunidades a identificar.

El problema de detección de comunidades (CDPs, por sus siglas en inglés) ha sido abordado por varios métodos, los cuales se agrupan fundamentalmente en dos categorías [4]: los métodos basados en heurísticas específicas [10,11], y aquellos basados en enfoque de optimización [12,13,14,15]. Uno de los métodos pioneros basado en el enfoque de heurística específica lo constituye el método jerárquico propuesto por [16], el cual divide iterativamente la red en comunidades, a partir de la identificación y eliminación del enlace de intermediación. Esta clase de métodos basados en una heurística específica, asumen o toman un conocimiento específico del problema para lograr una mayor eficacia. Si bien estos métodos resultan convenientes en muchos contextos aplicados, tienen la característica de ser muy dependientes del dominio de aplicación para el que fueron diseñados.

Por otro lado, los métodos basados en optimización siguen un enfoque más descriptivo. En este sentido, identifican las comunidades a partir de medir el valor óptimo de una función objetivo, que describe la comunidad como un conjunto de nodos con gran densidad de enlaces intra-grupo y baja densidad inter-grupos [10]. Esta propiedad, consistente en medir la densidad dentro de la comunidad, ha sido definida mediante diferentes funciones [17,18] y constituye el criterio de agrupamiento utilizado para identificar las comunidades. De esta forma, se han desarrollado métodos que buscan optimizar la función de *Modularidad* [11], otros que optimizan la función de *Corte* [19], así como aquellos que miden el valor óptimo de la función denominada *Longitud de la descripción* [20].

Si bien los algoritmos basados en optimización, han obtenido buenos resultados en una amplia gama de aplicaciones, el hecho de optimizar una sola función objetivo, hace que quede sesgada la estructura de comunidad a propiedades específicas [4]. O sea, se impone, explícitamente o no, un tipo de organización de los elementos de la red de acuerdo al criterio o función objetivo que se utilice. Por tanto, si las estructuras desconocidas de comunidades subyacentes en la red no presentan la propiedad que define la función a optimizar, no se obtendrán resultados satisfactorios. Un ejemplo de esto está dado con el problema del límite de resolución de la función Modularidad, la cual al ser optimizada, tiende a fallar en la identificación de comunidades más pequeñas que una escala que está determinada por el tamaño de la red y el grado de interconexión entre las mismas [21]. Otras investigaciones, tales como [22,23,17], han revelado este mismo problema en otras funciones objetivos.

De esta forma, el optimizar una única función objetivo, no permite cubrir potencialmente la variedad de características y/o estructuras presentes en los datos como por ejemplo, tamaño diverso de comunidades, formas convexas o no de las mismas, uniformidad, alta densidad intra-grupos y separabilidad inter-grupos, entre otras. Incluso, propiedades o características determinadas por un dominio de aplicación en particular. En este sentido, resulta complejo poder determinar a priori la función objetivo que mejor desempeño tenga en un problema en particular. A esto se suma que no existe un algoritmo de detección de comunidades que pueda ser utilizado en cualquier tipo de red con resultados satisfactorios, toda vez que, la naturaleza no supervisada del problema, las propiedades propias del mismo, las características de los objetos y sus relaciones, así como las particularidades del método de solución, complejizan su solución óptima.

Con base en el análisis anterior, se puede concluir que si se utiliza una variedad de criterios o funciones a optimizar simultáneamente, en principio se contaría con una complementariedad de enfoques. Este compromiso entre varias funciones ayudaría a identificar diferentes estructuras

subyacentes en el conjunto de objetos, lo cual mediante métodos tradicionales requerirá la aplicación por separado de algoritmos de diferentes enfoques. Este enfoque de abordar un problema, a partir de optimizar al unísono diferentes funciones, en ocasiones contradictorias, se conoce como optimización multi-objetivo (MOPs, por sus siglas en inglés) [3] y ha sido ampliamente abordado en muchos contextos aplicados [24]. En los últimos años, el problema de detección de comunidades basado en optimización se ha modelado mediante dicho enfoque y ha mostrado ventajas respecto al enfoque tradicional [1].

De hecho, algunos investigadores han descrito cómo la enumeración de las comunidades en una red constituye una solución de compromiso entre varios criterios. Por ejemplo, en [21] se enuncia cómo la obtención del valor máximo de la Modularidad constituye la búsqueda de un compromiso o equilibrio ideal entre el número de módulos y el índice de calidad en cada uno de ellos. De manera análoga, en [25] se describe cómo la Modularidad es la combinación de dos componentes en conflicto, consistentes en maximizar el número de enlaces intra-grupo junto con minimizar el grado total de enlaces por comunidad. También en [20], se hace la observación de que la enumeración de las comunidades en la red es inevitablemente un compromiso entre la cantidad de la información de la estructura de la red y su longitud de descripción. Si bien estos trabajos enuncian el compromiso intrínseco existente en la optimización de estas funciones, aún la mayoría de los algoritmos optimizan un solo objetivo, incluso en el caso que este combine componentes en conflictivo, tal y como ocurre con la función de Modularidad [11].

Desde el enfoque de optimización multi-objetivo, el problema de detección de comunidades en redes sociales, ha sido abordado por diferentes métodos en los últimos años [26,4,2,1]. En este sentido, se busca optimizar un conjunto de funciones objetivos que midan una o varias propiedades de las comunidades y de esta forma, obtener una o varias estructuras de grupos o agrupamientos, que describan una organización en comunidades de la red en estudio. En los MOPs, normalmente no existe una única solución al problema, dado que se tiene que optimizar varias funciones al unísono. De esta forma, un enfoque ampliamente utilizado es el conocido como *dominancia de Pareto*. Se dice que una solución domina a otra si no es peor en ninguna de las funciones a optimizar y en al menos una de ellas es mejor [3].

Por tanto, de manera general, el problema computacional a resolver sería encontrar el conjunto de soluciones no dominadas, denominado *Conjunto de Pareto* (PS, por sus siglas en inglés), que mejor compromiso logran en la optimización del conjunto de funciones objetivos. Este compromiso constituye la proyección del conjunto de Pareto en el vector de funciones objetivos y es conocido como *Frente de Pareto* (PF, por sus siglas en inglés) [3]. De esta forma, los agrupamientos encontrados muestran diferentes puntos de vista o enfoques de las estructuras en comunidades presentes en la red. En principio, esto constituye un apoyo a la obtención de estructuras complejas y a su mejor comprensión, toda vez que se midieron u optimizaron, mediante las funciones objetivos, diferentes propiedades deseables en las comunidades. Además de atenuar el sesgo causado por la medición de una única propiedad, la cual puede ser adecuada solo para un cierto tipo de redes [4].

Entre los aspectos fundamentales a contemplar en la solución del problema de detección de comunidades en redes sociales, a partir de un enfoque de optimización multi-objetivo, se encuentran: la definición de comunidad, la representación de la solución, las funciones objetivos, el método de solución del problema y la selección de la solución final. A la par con lo anterior, existen contextos del problema de detección de comunidades que condicionan y/o dificultad los aspectos anteriores, como es el caso de las comunidades descritas en entornos solapados o difusos, redes donde los enlaces entre los elementos son disímiles y con ponderaciones positivas y negativas

o en dominios de aplicación donde se cuenta con la descripción de los objetos de la red a partir de atributos o rasgos. Además, existen escenarios en donde la red puede presentar cambios a partir de la adición y/o eliminación de nodos y aristas.

En tal sentido, la mayoría los métodos propuestos desde el enfoque multi-objetivo tales como [2,4,26,1], solo abordan comunidades duras, o sea, aquellas donde los elementos de la red pertenecen solo a una comunidad. Esto no refleja con certeza la realidad presente en las redes sociales, donde es común que los elementos pertenezcan a diferentes comunidades. Así mismo, pocos son los trabajos [27,28] que abordan ponderaciones en los enlaces, en estos casos positivas y negativas. Estas ponderaciones dan una mayor descripción semántica a las relaciones existentes entre los elementos de la red social.

El trabajo de [29], constituye la única propuesta donde los elementos de la red poseen diferentes tipos de enlaces. Esto permite modelar el fenómeno donde los objetos de la red están conectados por diferentes tipos de relaciones [30], como por ejemplo, trabajo, amistad y familia, entre otras. En cuanto al procesamiento de cambios en la red, solo los trabajos de [31,32,33] abordan este aspecto. En la actualidad, resulta muy común en el contexto de las redes sociales evidenciar cambios, en términos de adiciones y/o eliminaciones de vértices y aristas.

Con base en lo planteado anteriormente, se puede concluir que en el contexto del descubrimiento de comunidades siguiendo un enfoque de optimización multi-objetivo persisten varias limitaciones: (1) pocos son los métodos que abordan la detección de comunidades solapadas o difusas, (2) la mayoría de los trabajos no contemplan los pesos en los enlaces, existen un pequeño número que contemplan los pesos positivos y negativos, (3) solo se evidencia un trabajo sobre redes caracterizadas por diferentes tipos de enlaces entre sus elementos, (4) el tratamiento a los cambios que puede sufrir la red, ha sido abordado en mucha menor medida que el contexto estático. Las insuficiencias detectadas permiten afirmar que el problema de detección de comunidades desde un enfoque multi-objetivo sigue siendo un área novedosa y pertinente.

Por lo antes expuesto, en este trabajo se mostrarán de manera sucinta las bases conceptuales sobre las que descansan los principales métodos de detección de comunidades con un enfoque multi-objetivo, así como se identificarán las principales insuficiencias de estos y los posibles trabajos futuros a desarrollar sobre esta temática. De esta forma, en la sección 2 se discutirán los aspectos fundamentales relacionados con las comunidades, sus definiciones, así como los conceptos principales relacionados con los problemas de optimización multi-objetivo. Además se dará una descripción detallada de las funciones objetivos principales utilizadas como instrumento para medir o guiar el proceso de búsqueda desarrollado por los algoritmos estudiados.

Una vez definido el problema computacional a abordar, se describirán los aspectos fundamentales de los métodos de solución en la sección 3, en donde se realizará un análisis de las representaciones de solución, los métodos y operadores heurísticos fundamentales utilizados. La descripción de los métodos fundamentales pertenecientes al estado del arte, así como sus principales limitantes se encuentran agrupadas por tipos de problemas. Estos tipos de problemas están orientados a detectar comunidades en redes estáticas, redes con signo, redes que procesan cambios y las redes que presentan diferentes tipos de enlaces, denominadas redes multi-características. A esto se dedicará la sección 4. Por último, en la sección 5, se plantean las conclusiones.

2. Marco teórico

En la presente sección se abordan los conceptos y definiciones fundamentales utilizadas a lo largo de la investigación.

2.1. Definición del problema

Sea G una red no dirigida, modelada por un grafo $G(V, E)$, donde V ($|V| = n$) y E ($|E| = m$) representan el conjunto de vértices y aristas, respectivamente. El problema de detección de comunidades puede considerarse como la construcción o identificación de una partición $C = \{C_1, C_2, \dots, C_k\}$ compuesta por k grupos o comunidades. Existen fundamentalmente dos clases de enfoques de solución al problema anterior, los basados en heurísticas específicas y los basados en optimización. Ambos casos pueden considerarse como un problema de optimización mono-objetivo (Ω, F) , el cual se define a continuación [4].

Definición 1. Detección de comunidades desde un enfoque mono-objetivo. Determinar la partición C^* tal que:

$$F(C^*) = \min_{C \in \Omega} F(C), \quad (1)$$

donde Ω representa el conjunto de soluciones válidas, C la estructura de comunidad de la red G y $F : \Omega \rightarrow R$ una función objetivo. Esta última asigna un valor de aptitud a una solución $C \in \Omega$, lo cual permite establecer una relación de orden total entre cualquier par de soluciones pertenecientes al espacio de solución Ω [34].

Como se argumentó en la sección anterior, el enfoque mono-objetivo dificulta la identificación o construcción de las posibles diversas estructuras presentes en una red. Una de las alternativas estudiadas en los últimos años consiste en modelar la detección de comunidades como un problema de optimización multi-objetivo $(\Omega, F_1, F_2, \dots, F_q)$, es decir, donde se tiene más de una función a optimizar. Este problema se define de la siguiente forma [4].

Definición 2. Detección de comunidades desde un enfoque multi-objetivo. Determinar la partición C^* tal que:

$$F(C^*) = \min_{C \in \Omega} (F_1(C), F_2(C), \dots, F_q(C)), \quad (2)$$

donde q constituye el número de funciones objetivos a optimizar y F_i la i -ésima función. Se asume sin pérdida de generalidad que todas las funciones se minimizan. Al optimizar múltiples funciones, usualmente no se cuenta con una única solución óptima del problema, sino con un conjunto de soluciones. Uno de los conceptos ampliamente utilizado en este tipo de problema lo constituye la optimalidad de Pareto, el cual se define de la siguiente forma [3].

Definición 3. Dominancia de Pareto. Dadas dos estructuras de comunidades $C_1, C_2 \in \Omega$, se dice que C_1 domina a C_2 , denotándose como $C_1 \preceq C_2$, si y solo si:

$$\forall i \in \{1, \dots, q\} \ F_i(C_1) \leq F_i(C_2) \ \wedge \ \exists i \in \{1, \dots, q\} : F_i(C_1) < F_i(C_2), \quad (3)$$

o sea, una estructura de comunidad o agrupamiento domina a otro si no es peor en ninguna de las funciones objetivos a optimizar y en al menos una de ellas es mejor. A partir de la relación de orden parcial definida anteriormente, se define el óptimo de Pareto de la siguiente forma [3].

Definición 4. Óptimo de Pareto. Una estructura de comunidad $C^* \in \Omega$, se dice que es un óptimo de Pareto si y solo si:

$$\forall C \in \Omega, C \not\preceq C^*, \quad (4)$$

es decir, no existe otra solución perteneciente al conjunto de soluciones válidas que la domine. El hecho de que existan varias funciones a optimizar, en ocasiones contradictorias, hace que existan un conjunto de soluciones que cumplan ser óptimos de Pareto. Por tanto, se define el conjunto de soluciones no dominadas o conjunto de Pareto de la siguiente forma [3].

Definición 5. Conjunto de Pareto. Las estructuras de comunidad pertenecientes al conjunto de soluciones válidas Ω constituyen el Conjunto de Pareto si y solo si:

$$PS = \{C \in \Omega : \nexists C' \in \Omega, C' \preceq C\}, \quad (5)$$

o sea, no existe una solución del conjunto de soluciones válidas que la domine. Por último, estas soluciones evaluadas en el conjunto de funciones objetivos constituyen la frontera de Pareto y se define de la siguiente forma [3].

Definición 6. Frontera de Pareto. Es el conjunto de la proyección de las soluciones pertenecientes al Conjunto de Pareto:

$$FS = \{F(C) : C \in PS\}. \quad (6)$$

En este sentido, el conjunto de funciones objetivos, $F = \{F_1, F_2, \dots, F_q\}$, puede estar constituido por las funciones ampliamente utilizadas en la detección de comunidades como Modularidad [11], función de corte [19] o longitud de la descripción [20], así como cualquiera de las descritas en [17]. Incluso, se pueden definir nuevas funciones objetivos en dependencia de contextos específicos.

2.2. Definición del concepto de comunidad

Aún cuando no existe un consenso en cuanto al concepto de comunidad, intuitivamente, se desea que la densidad de enlaces dentro de la comunidad sea superior a los existentes fuera de esta. Aunque en [6] se dan diferentes características acerca de comunidad, es en el trabajo de [10] donde se da un concepto formal de comunidad en función de la densidad de enlaces internos respecto a los externos.

El concepto básico a contemplar lo constituye el grado de un vértice v_i , el cual está definido a partir de la matriz de adyacencia A del grafo G como: $k_{v_i} = \sum_{v_j \in V} A_{v_i, v_j}$. La matriz de adyacencia representa la topología de la red; es decir, su información estructural. En un grafo no dirigido y no ponderado, una celda de esta matriz toma valor 1 si existe el enlace entre los vértices v_i y v_j , en otro caso, toma valor cero. Si se considera un subgrafo $S \subset G$ al cual pertenece v_i , el grado total de v_i puede descomponerse en dos contribuciones: $k_{v_i}(S) = k_{v_i}^{in}(S) + k_{v_i}^{out}(S)$. $k_{v_i}^{in}(S) = \sum_{v_j \in S} A_{v_i, v_j}$ representa el número de enlaces que tiene el nodo v_i con el resto de elementos de la comunidad S , denominándose grado interno del vértice v_i respecto a S . Por el contrario, $k_{v_i}^{out}(S) = \sum_{v_j \notin S} A_{v_i, v_j}$ constituye el número de enlaces que tiene v_i con el exterior de la comunidad S y se conoce como grado externo del vértice v_i respecto a S . A partir de lo anterior, [10] define los conceptos de comunidad *fuerte* y comunidad *débil*.

Definición 7. Comunidad fuerte. Un subgrafo o comunidad S constituye una comunidad fuerte si y solo si:

$$k_{v_i}^{in}(S) > k_{v_i}^{out}(S), \quad \forall v_i \in S. \quad (7)$$

Una comunidad se considera como *fuerte* si todos los vértices que pertenecen a la misma tienen más enlaces dentro de la comunidad que fuera de esta.

Definición 8. Comunidad débil. El subgrafo o comunidad S constituye una comunidad débil si y solo si:

$$\sum_{i \in S} k_i^{in}(S) > \sum_{i \in S} k_i^{out}(S). \quad (8)$$

Una comunidad se considera como *débil* si se cumple que, entre todos los vértices de la comunidad, hay más enlaces con vértices dentro de la comunidad que con vértices fuera de esta. Se evidencia que una comunidad fuerte es a la vez débil, no ocurriendo lo contrario.

2.3. Funciones objetivos en la detección de comunidades

En los trabajos de [23,17] y [18] se da una descripción de las funciones objetivos más utilizadas en la detección de comunidades. Cabe destacar que no todas las funciones descritas en [17] son utilizadas por los métodos CDPs multi-objetivos. No obstante, estas funciones describen propiedades útiles a medir en las comunidades como son la densidad intra-grupo, dispersión inter-grupo, etc. Por tanto, se han definido funciones objetivos que miden estas propiedades en el contexto del CDPs con enfoque multi-objetivo. A continuación se describen las funciones utilizadas en los algoritmos multi-objetivo para la detección de comunidades.

2.3.1. Funciones basadas en la Modularidad

Muchos de los trabajos reportados que utilizan un enfoque multi-objetivo [35,31,32], utilizan la medida de Modularidad, una transformación de esta [26] o las componentes definidas en la misma [4]. La presente sección agrupa este tipo de funciones objetivos.

Definición 9. Modularidad [11]. Dado el grafo $G = (V, E)$, la función de Modularidad se define como:

$$Q(C) = \sum_{i=1}^k \left[\frac{|E(C_i)|}{m} - \left(\frac{\sum_{v \in C_i} deg(v)}{2m} \right)^2 \right], \quad (9)$$

donde $|E(C_i)|$ representa el número de enlaces en la comunidad C_i , m el número de enlaces en la red, C constituye un agrupamiento o un conjunto de comunidades y $deg(v)$ el grado del vértice v perteneciente a la comunidad C_i .

Definición 10. Modularidad intra-grupo [4]. La función Modularidad intra-grupo mide densidad intra-grupo y queda definido de la siguiente forma:

$$intra(C) = 1 - \sum_{i=1}^k \frac{|E(C_i)|}{m}. \quad (10)$$

Definición 11. Modularidad inter-grupo [4]. La función Modularidad inter-grupo mide las conexiones inter-grupo y queda definido de la siguiente forma:

$$inter(C) = \sum_{i=1}^k \left(\frac{\sum_{v \in C_i} deg(v)}{2m} \right)^2. \quad (11)$$

Cabe señalar que $intra(C)$ y $inter(C)$ constituyen el primer y segundo término de la función Modularidad, respectivamente. Así mismo, constituyen, tal y como se plantea en [25] y [21], un compromiso entre dos aspectos fundamentales a medir en la comunidad. Son estos, alta densidad intra-grupo y baja conexión inter-grupo [10].

Se han propuesto varias modificaciones de la función Modularidad en diferentes contextos [36,37,38,39]. Una de las modificaciones se denomina *densidad de modularidad* [37], la cual se define a continuación.

Definición 12. Densidad de la modularidad [37].

$$D = \sum_{i=1}^k \frac{L(V_i, V_i) - L(V_i, \bar{V}_i)}{|V_i|}, \quad (12)$$

donde $V_i \subseteq V$ y $\bar{V}_i = V - V_i$ representan los vértices internos y externos de la comunidad C_i , respectivamente y $L(V_i, V_i) = \sum_{i,j \in V_i} A_{ij}$, $L(V_i, \bar{V}_i) = \sum_{i \in V_i, j \in \bar{V}_i} A_{ij}$ constituyen el número de enlaces internos y externos de la comunidad, respectivamente.

El primer término de la función Densidad de la modularidad, definido en [40] y denominado *Ratio Association (RA)*, representa la densidad intra-grupo la cual se quiere maximizar. A partir de posteriores modificaciones de esta medida surgieron las medidas *Negative Ratio Association (NRA)* [26] y *Modified Ratio Association (MRA)* [41]. Esta última medida ha sido referenciada también como *Kernel K-means (KKM)* en el trabajo [27]. El *Ratio Cut (RC)*, busca minimizar las conexiones de la comunidad con el exterior [42] y coincide con el segundo término de la función densidad de la modularidad.

Como se puede observar, tanto la Modularidad como la Densidad de la modularidad representan un compromiso entre la densidad intra-grupo y los enlaces de las comunidades con el exterior; es decir, la densidad intra-grupo trata de maximizar los enlaces internos de los vértices pertenecientes a la comunidad, a la vez que se trata de minimizar los enlaces de estos vértices con los vértices externos a la comunidad. A continuación se definen formalmente las medidas descritas anteriormente.

Definición 13. Ratio Association [40].

$$RA = \sum_{i=1}^k \frac{L(V_i, V_i)}{|V_i|}. \quad (13)$$

Esta función es decreciente respecto al número de comunidades [26]; es decir, en la medida que crece el número de comunidades decrece el valor de la función y viceversa.

Definición 14. Negative Ratio Association [26].

$$NRA = - \sum_{i=1}^k \frac{L(V_i, V_i)}{|V_i|}. \quad (14)$$

Definición 15. Modified Ratio Association [41].

$$MRA = 2(n - k) - \sum_{i=1}^k \frac{L(V_i, V_i)}{|V_i|}, \quad (15)$$

donde n y k representan el número de vértices y el número de comunidades respectivamente.

Definición 16. Ratio cut [42].

$$RC = \sum_{i=1}^k \frac{L(V_i, \bar{V}_i)}{|V_i|}. \quad (16)$$

En este caso, a diferencia de la función RA, RC es una función creciente respecto al número de comunidades; es decir, en la medida que aumenta el número de comunidades la función toma mayor valor y en tal sentido mayor número de enlaces externos contiene la comunidad. Lo anterior muestra la contradicción entre ambas medidas [26].

2.3.2. Funciones basadas en la teoría de grafo

Muchas de las funciones utilizadas en la detección de comunidades se basan en la teoría de grafo [23,17]. Estos criterios definen relaciones que pueden involucrar tanto los enlaces internos, aquellos enlaces que involucran solo los vértices pertenecientes a la comunidad, como externos a la comunidad, son los enlaces que involucran un vértice de la comunidad con otro externo a esta. También utilizan el número de vértices y/o enlaces tanto de la comunidad como de la red en su totalidad. A continuación se describen y definen algunas de las funciones basadas en grafo, utilizadas en la detección de comunidades desde el enfoque multi-objetivo.

Definición 17. Community fitness (CF) [43]. Sea $S \subset G$ un subgrafo que representa una comunidad o grupo y $k_{v_i}^{in}(S)$ y $k_{v_i}^{out}(S)$ los grados internos y externos del nodo v_i , respectivamente. CF queda definida como:

$$\sum_{i=1}^k P(S_i), \quad (17)$$

siendo $P(S)$:

$$P(S) = \sum_{i \in S} \frac{k_i^{in}(S)}{(k_i^{in}(S) + k_i^{out}(S))^\alpha}, \quad (18)$$

donde α , denominado parámetro de resolución, representa un valor positivo que controla el tamaño de la comunidad.

Definición 18. Community score (CS) [2]. Sea $\mu_i = \frac{1}{|S|} k_i^{in}(S)$ la relación promedio entre el grado interno del vértice i y la cardinalidad de la comunidad S . Sea $M(S) = \frac{\sum_{i \in S} (\mu_i)^r}{|S|}$ la acumulación promedio de la relación anterior por cada vértice perteneciente a la comunidad, donde $0 \leq \mu \leq 1$. El volumen de la comunidad S se define como $v_S = \sum_{i,j \in S} A_{ij}$ y representa el total de enlaces internos en la comunidad. A partir de $M(S)$ y v_S se define $score(S) = M(S) \times v_S$, lo cual contempla tanto los enlaces entre los nodos así como el total de enlaces de la comunidad.

La función objetivo CS queda definida como:

$$CS(S) = \sum_{i=1}^k score(S_i), \quad (19)$$

donde $\{S_1, S_2, \dots, S_k\}$ representa una estructura de comunidades.

Definición 19. MinMaxCut (MMC) [44].

$$MMC = \sum_{i=1}^k \frac{d^{out}(V_i)}{d^{in}(V_i)}, \quad (20)$$

donde $d^{in}(V_i) = \sum_{j \neq k, j \in V_i, k \in V_i} A_{jk}$ y $d^{out}(V_i) = \sum_{j \neq k, j \in V_i, k \notin V_i} A_{jk}$ representan el número de enlaces entre los vértices pertenecientes a la misma comunidad y el número de conexiones de estos, con vértices no pertenecientes al grupo, respectivamente.

2.3.3. Aspectos a destacar en las funciones objetivos

Problema del límite de resolución

Como se ha explicado con anterioridad, no existe consenso en qué propiedades debe cumplir un subgrafo de la red para ser considerado una comunidad. Sin embargo, se tiene mayor claridad en qué tipo de grafos no tienen una estructuración basada en comunidades. La existencia de dichos grupos en una red, indica que existen grupos de vértices que tienden a interactuar más entre ellos que con el resto de los nodos. Por tal motivo, en una red donde la probabilidad de que un par de vértices estén conectados es la misma para todos, se considera que no existen comunidades. Los grafos aleatorios son un ejemplo de este tipo de redes y en el contexto de detección de comunidades este tipo de grafos se utiliza mayormente para definir redes que no se organizan en grupos. Este tipo de grafo son conocidos como modelos de grafo aleatorio nulo y deben compartir alguna propiedad estructural con respecto al grafo G como es tener la misma distribución de grados de los vértices que G [45].

A partir de lo anterior, se puede concluir que cualquier estructuración o partición del modelo de grafo aleatorio nulo tendrá más enlaces entre vértices pertenecientes a comunidades distintas que entre vértices de la misma comunidad. De esta forma, una red cuya estructura subyacente está basada en comunidades, difiere notablemente de sus versiones aleatorias [45]. La idea anterior, constituye el sustento básico de la medida de Modularidad. Esta medida calcula la diferencia existente entre la distribución de enlaces de la red original y su modelo de grafo aleatorio nulo. Una característica útil de la función de Modularidad es que no mide solamente la eficacia de la partición mediante valores positivos, sino que también puede describir la ineficiencia de la comunidad mediante valores negativos. Por tanto, una partición con valores negativos implica la existencia de grupos de baja densidad interna que interactúan mucho más con el resto de la red [45].

No obstante, en [21] se hace un análisis del comportamiento de la medida de Modularidad para determinar estructuras de cierto tamaño. La función de Modularidad, al basarse en un modelo de grafo aleatorio nulo, supone que todos los pares de vértices pueden estar conectados con la misma probabilidad. Este supuesto resulta poco práctico debido a que en redes de gran tamaño, los vértices de una comunidad limitan sus enlaces a una parte de la red. De esta forma, la Modularidad queda expresada en función del número de enlaces total existente en la red. Por tanto,

como se demuestra en [21], esta medida, que establece cuándo un grupo de vértices constituye una comunidad a partir de sus enlaces, se encuentra acotada por el número total de enlaces de la red. De esta forma, contradice la idea intuitiva de que los vértices pertenecientes a una comunidad deben tener sus enlaces mayormente con vértices de una parte de la red.

Correlación entre las funciones objetivos a optimizar

En el trabajo de [46] se realiza un análisis de la importancia que tiene para el agrupamiento multi-objetivo, el tipo de correlación existente entre las funciones a optimizar. Este análisis se realiza mediante un caso de estudio para ilustrar cómo existen agrupamientos que resulta complejo identificar mediante la optimización de una sola función objetivo.

En el contexto de la detección de comunidades, en los trabajos descritos en [23,47,17] se realiza un análisis con mayor rigor respecto a la correlación que debe prevalecer entre las funciones objetivos utilizadas en la detección de comunidades para identificar estructuras más eficaces. En este sentido, se describe cómo aquellas funciones que están correlacionadas negativamente son las más convenientes a utilizar para identificar estructuras de comunidades eficaces. En el caso de los métodos que utilizan funciones que están correlacionadas de manera positiva, son los que brindan resultados menos satisfactorios.

Según los autores del trabajo [17], las funciones objetivo correlacionadas negativamente tienen un efecto inverso respecto al número de comunidades. Esto permite que se obtengan estructuras donde varía el número de comunidades de la red y de esta forma, se evita que se encuentren soluciones triviales; e.g. comunidades de un solo elemento o una sola comunidad que involucre todos los vértices de la red. Además, este tipo de correlación ayuda a identificar estructuras con diferente granularidad y por tanto, diferentes niveles de resolución, lo cual en principio contribuye a un mejor análisis de la red. En el caso de las funciones correlacionadas de manera positiva, resulta similar a optimizar un solo objetivo [3] y por tanto, el algoritmo tendría un enfoque mono-objetivo. Por último, en el caso de los objetivos que no están correlacionados, donde la contradicción entre ellos no es fuerte, resulta complejo recorrer el espacio de búsqueda y evitar los óptimos locales.

No todas las funciones descritas en la sección 2.3 son incluidas en estos trabajos. Por tanto, es necesario realizar un análisis con mayor rigor para identificar la combinaciones de funciones objetivos a utilizar en el proceso de búsqueda que ayude a identificar estructuras con mayor eficacia.

3. Métodos de solución de problemas

Tal y como se definió en la sección 2.1, un problema de optimización se modela como una tupla (Ω, F) , en donde Ω representa el espacio de solución del problema y F una función objetivo que asigna un valor de *aptitud* a una solución válida del espacio de solución. Para abordar este tipo de problemas existen distintas clases de métodos, como los métodos exactos y los métodos de aproximación [34]. Los métodos exactos, entre los que podemos mencionar, los basados en programación dinámica, los basados en ramificación y poda, así como los métodos A^* , dan garantías de obtener el óptimo global, aunque para problemas computacionalmente complejos, como aquellos pertenecientes a la clase NP-completo, no resultan viables por su costo computacional. Para abordar este clase de problemas computacionalmente complejos, se han desarrollado métodos de aproximación que no dan garantías de óptimo global pero permiten la obtención de buenas soluciones, con un costo computacional bajo.

Según [34], los métodos aproximados o métodos no exactos, se pueden clasificar en: (1) algoritmos aproximados y (2) métodos heurísticos. Los algoritmos aproximados permiten, bajo ciertas acotaciones y/o supuestos del problema, identificar una solución que se encuentra a cierta distancia del óptimo global. Sin embargo, estos métodos son dependientes del problema en cuestión, lo cual limita su aplicabilidad. Por otra parte, en la práctica, las aproximaciones alcanzables están demasiado lejos de la solución óptima global, por lo que estos algoritmos no son muy útiles para muchas aplicaciones de la vida real [34]. En el caso de los métodos heurísticos se clasifican en métodos de heurística específica y metaheurísticas. Los métodos de heurística específica, hacen uso de un conocimiento particular del problema para definir estrategias de búsqueda que identifiquen en un tiempo razonable buenas soluciones. Son métodos dependientes del problema, lo cual dificulta generalizar su aplicación; sin embargo, son algoritmos muy eficaces en el contexto del problema para el cual fueron diseñados.

En el caso de las metaheurísticas, proponen de manera general un marco de trabajo que sirve de guía para diseñar algoritmos de optimización. Este marco de trabajo incluye fundamentalmente el diseño y/o formulación de: (a) una representación del problema, (b) la definición del espacio de estados del mismo, (c) la función o funciones objetivos que guían el proceso de búsqueda, (d) así como el diseño de operadores heurísticos que permiten construir y/o transitar por el espacio de búsqueda del problema. De esta forma, se permite diseñar heurísticas específicas para un dominio de aplicación en particular [34].

Las metaheurísticas pueden clasificarse por diferentes aspectos. En este sentido, según su inspiración se clasifican en métodos basados en la naturaleza o basados en fenómenos físicos [34]. Entre los basados en la naturaleza, está la clase de *Algoritmos Evolutivos* (EA, por sus siglas en inglés). Esta contempla los *Algoritmos Genéticos* (GAs, por sus siglas en inglés), las *Estrategias Evolutivas* (ES, por sus siglas en inglés), la *Programación Genética* (GP, por sus siglas en inglés) y los *Algoritmos de Estimación de Distribuciones* (EDA, por sus siglas en inglés), entre otros. Otros tipos de métodos basados en la naturaleza lo constituye la clase conocida como *Enjambre Inteligente* (SI, por sus siglas en inglés), la cual contempla entre sus métodos más utilizados la *Optimización basada en Enjambre de Partículas* (PSO, por sus siglas en inglés) y la *Optimización basada en Colonia de Hormigas* (ACO, por sus siglas en inglés).

Por otro lado, según el número de soluciones o espacios de estados que contemplan, los métodos son clasificados según [34] en poblacionales o de trayectoria simple. Entre los métodos poblacionales se encuentran, PSO, ACO y EA, comentados anteriormente. Estos métodos se caracterizan por trabajar con una población de soluciones o espacios de estados del problema. Por el contrario, los métodos basados en trayectoria simple trabajan solo con un estado del problema. Ejemplos de estos últimos métodos son *Escalador de Colina* (HC, por sus siglas en inglés), *Recocido Simulado* (SA, por sus siglas en inglés) y *Búsqueda Tabú* (TS, por sus siglas en inglés).

De manera general, los métodos poblacionales tienden a lograr una mayor exploración del espacio de solución del problema durante la búsqueda, al utilizar un conjunto de soluciones o espacios de estados del problema. Por el contrario, los métodos basados en trayectoria simple tienden a obtener, en principio, una mayor explotación del espacio de solución del problema. Esto último permite una convergencia rápida a un óptimo local. Lograr un adecuado compromiso o balance entre exploración y explotación, constituye un reto en el uso de los algoritmos metaheurísticos en un dominio de aplicación concreto [3,34].

En los problemas de optimización multi-objetivos, no se cuenta frecuentemente con una única solución óptima del problema, esto se debe fundamentalmente a que los q criterios o funciones objetivos a optimizar (F_1, F_2, \dots, F_q) están en conflicto. Por tanto, se define una relación de orden

parcial, conocida como relación de dominancia y definida en 2.1, entre las diferentes soluciones del problema en términos de su valor en las funciones objetivos [3,34]. Con base en la relación definida anteriormente, el problema original se puede transformar en encontrar el conjunto de soluciones no dominadas o conjunto de Pareto definido en 2.1. Este conjunto incluye las soluciones que logran un compromiso entre las diferentes funciones objetivos en conflicto. El problema de encontrar el conjunto de Pareto en el contexto del problema de detección de comunidades ha sido abordado mediante metaheurísticas.

En el caso del problema de detección de comunidades desde un enfoque multi-objetivo [35,2,4], las metaheurísticas más utilizadas son las conocidas como *Algoritmos Evolutivos Multi-objetivos* (MOEAs, por sus siglas en inglés) [24]. Esta clase de métodos tiene sus principios en la teoría de la evolución natural. De manera general, se cuenta con una representación de la solución del problema, denominado codificación o cromosoma, así como con operadores evolutivos tales como cruzamiento y mutación, que funcionan generalmente mediante reglas estocásticas. Existen modificaciones de los MOEAs, entre las que caben destacar los MOEAs basados en Descomposición [48] y los MOEAs Interactivos [49]. Para una descripción detallada de los diferentes enfoques de esta clase de métodos revisar [24].

Cabe destacar que los MOEAs no son la única clase de métodos de solución de problemas en el contexto multi-objetivo. Las metaheurísticas de Enjambre Inteligente y de Trayectoria Simple han sido utilizadas en problemas multi-objetivos [50]. Concretamente, en la detección de comunidades desde un enfoque de optimización multi-objetivo, se han utilizado metaheurísticas de Trayectoria Simple [1], así como *Metaheurísticas Híbridas* [51,52,53]. Estas últimas fusionan diferentes metaheurísticas.

La detección de comunidades desde un enfoque multi-objetivo guarda una estrecha relación con el problema de agrupamiento multi-objetivo. En [54] se plantea un marco de trabajo general para el agrupamiento multi-objetivo, a partir del uso de los MOEAs. De esta forma, cualquier algoritmo de agrupamiento multi-objetivo que utilice un MOEAs debe contemplar entre sus pasos [54]:

1. Diseñar o escoger una representación de la solución y generar una población inicial de estas.
2. Escoger o diseñar el conjunto de funciones objetivos a utilizar en la identificación, medición de propiedades de los grupos.
3. Diseñar o escoger los operadores heurísticos, operadores evolutivos, para trabajar con la representación de la solución y permitir transitar por el espacio de solución del problema.
4. Diseñar o utilizar un adecuado procedimiento para mantener y mejorar el conjunto de soluciones no dominadas; i.e., conjunto de Pareto.
5. Desarrollar un método para la selección de una solución final del conjunto de Pareto.

Cabe resaltar que si bien estos pasos fueron propuestos para el agrupamiento multi-objetivo a partir del uso de los MOEAs [54], son en principio generalizables para cualquier metaheurística utilizada. Esto se logra a partir del diseño de operadores heurísticos, evolutivos en el caso de los MOEAs, que permitan la construcción y/o recorrido por los espacios de solución del problema abordado. Estos operadores pueden estar basados en reglas estocásticas, deterministas o una combinación de ambas [34]. A continuación se describen algunos tipos de representación utilizadas en el CDPs multi-objetivo.

3.1. Representación de la solución

Para dar solución al problema de detección de comunidades desde un enfoque de optimización, se necesita un formalismo para representar la solución al problema. Esta representación constituye potencialmente una posible solución al problema, la cual es evaluada mediante la función objetivo para determinar su aptitud [3].

En el trabajo de [54] se realiza una descripción detallada de las clases de representación utilizadas en el problema del agrupamiento multi-objetivo. Muchas de estas representaciones pueden ser utilizadas en el problema de detección de comunidades debido a su similitud con el problema de agrupamiento. Una de las representaciones es la basada en *punto*, la cual incluye la representación basada en etiqueta y la representación basada en localización adyacente [54]. A continuación se describen las características fundamentales de estas representaciones, siendo las más utilizadas en el problema de detección de comunidades desde un enfoque multi-objetivo.

3.1.1. Representación basada en etiqueta

La representación basada en etiqueta, constituye un vector de tamaño igual al número de objetos del problema. Cada posición denota la etiqueta del grupo al que pertenece el objeto representado en esa posición. De esta forma, si en la posición i se asigna un valor k , esto representa la asignación del objeto i -ésimo al grupo k -ésimo. Por tanto, el vector toma valores enteros del conjunto $\{1, 2, \dots, K\}$, siendo K el número de grupos o comunidades [55]. En la figura 1 se muestra un ejemplo de esta representación.

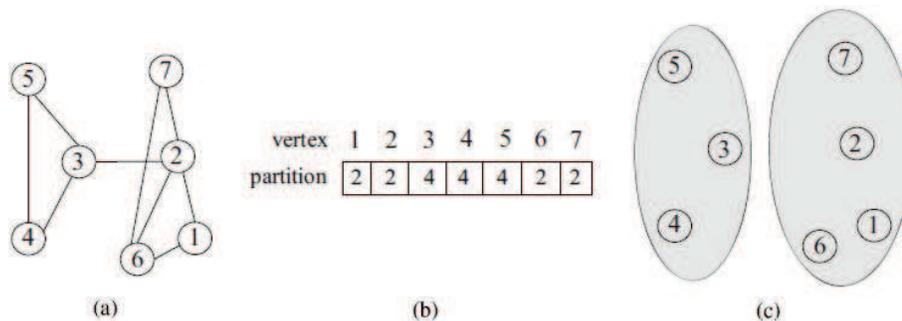


Fig. 1. Ejemplo de la representación basada en etiqueta: (a) topología del grafo, (b) representación de la solución, (c) estructura de comunidades [1].

Entre las principales ventajas de esta representación está la simplicidad de la construcción de los grupos o comunidades, debido a la asignación del grupo al que pertenece cada objeto. Así mismo, esta representación permite la identificación de grupos de forma arbitraria. Otra de las ventajas consiste en permitir trabajar con datos de cualquier tipo, tales como continuos y categóricos. Además, es independiente del número de grupos a identificar.

No obstante, entre sus limitaciones se encuentra su complejidad espacial, dado que el tamaño del vector está en función del número de objetos del problema. Esta limitación es importante en un contexto como la detección de comunidades en redes sociales, debido al tamaño de la red. Así mismo, tiene la limitante de representar soluciones redundantes. Por ejemplo, el vector 11122333 y 22233111 identifican el mismo agrupamiento o estructura de comunidades [55]. Finalmente, esta representación no permite formar grupos solapados, debido a que cada objeto es etiquetado y por ende, asignado a un único grupo [54]. Existen trabajos del estado del arte que

utilizan este tipo de representación [1,27,28], aún cuando la mayor parte utiliza la representación basada en localización adyacente.

3.1.2. Representación basada en localización adyacente

En los trabajos del estado del arte, resalta la utilización de la representación basada en grafo [55] denominada *representación basada en localización adyacente* [56]. Esta representación constituye un vector de dimensión n , siendo n el número de vértices del grafo, que contiene en cada posición un valor entero del conjunto $\{1, \dots, n\}$. De esta forma, si la posición i -ésima del vector contiene el valor j -ésimo, se representa el enlace que existe entre los vértices v_i y v_j . Por tanto, en el agrupamiento o estructura de comunidad resultante, ambos vértices pertenecen al mismo grupo o comunidad. Esta estructura de comunidad se obtiene a partir de la identificación de todas las componentes conexas del grafo [46]. El número de componentes conexas representa el número de comunidades o grupos. Por tanto, esta representación contempla tanto el agrupamiento como el número de grupos, el cual se determina de manera automática a partir de la identificación de las componentes conexas. La determinación automática del número de grupos o comunidades resulta muy útil en un contexto no supervisado como la detección de comunidades. La figura 2 muestra un ejemplo de esta representación.

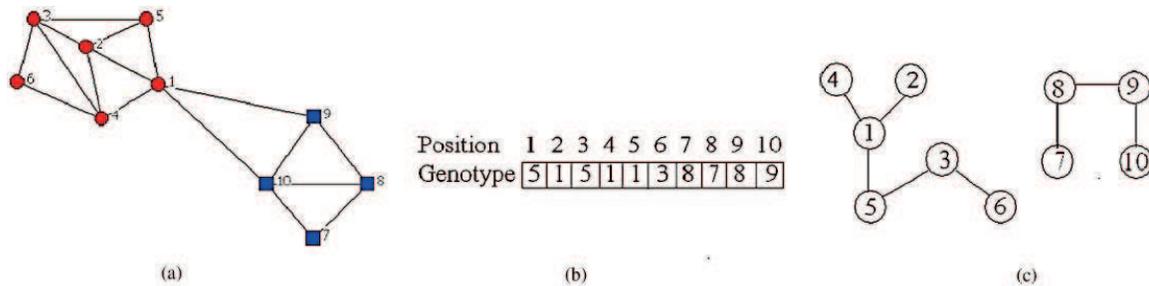


Fig. 2. Ejemplo de la representación basada en localización adyacente: (a) topología del grafo, (b) representación de la solución, (c) estructura de comunidades [2].

La representación basada en localización adyacente tiene entre sus ventajas, la posibilidad de determinar automáticamente el número de grupos, así como contemplar toda la estructura de comunidad en la representación. Similar a la representación basada en etiqueta, no condiciona la obtención de comunidades de forma esférica, lo cual permite detectar grupos de forma arbitraria [54]. Entre sus limitantes se encuentra su complejidad espacial, resultado de ser una representación de dimensión igual al número de objetos de la red. La mayoría de los trabajos del estado del arte utilizan este tipo de representación tal y como se muestra en la figura 3.

3.2. Algoritmos metaheurísticos

Como se describió en la sección 3, muchos de los problemas de optimización multi-objetivo y particularmente, en la detección de comunidades, han sido abordados utilizando los MOEAs. Una descripción detallada de los diferentes enfoques de solución a un MOPs se encuentra en [3], así como una explicación detallada de los diferentes tipos de MOEAs se encuentra en [24]. Así mismo, [50] describe las metaheurísticas no basadas en enfoque evolutivo, tales como PSO, ACO, HC y SA, entre otras.

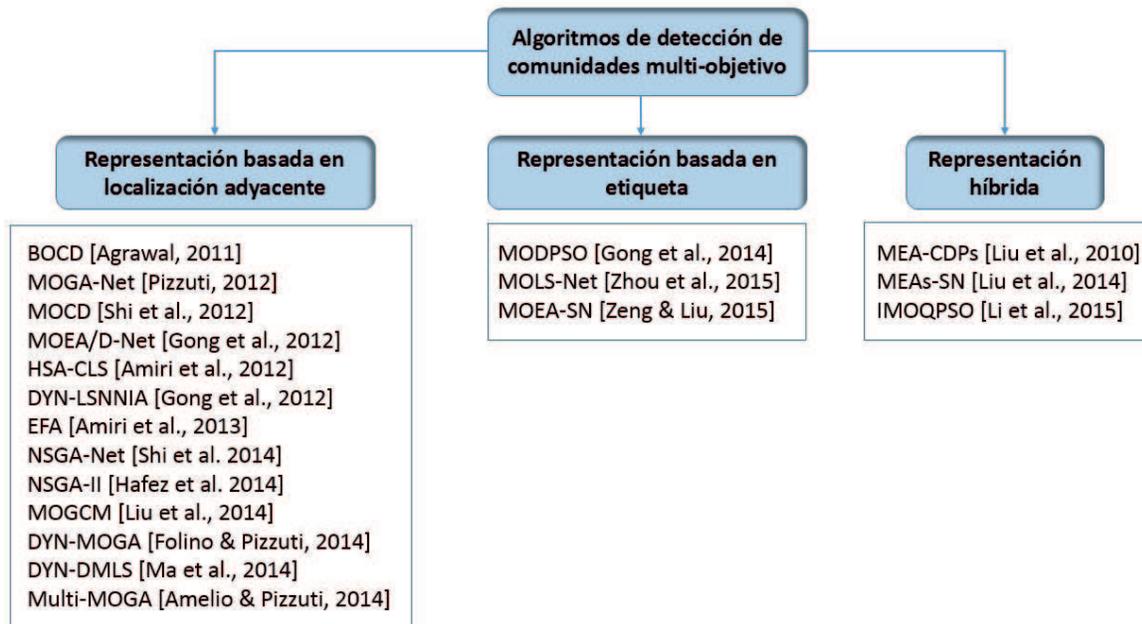


Fig. 3. Algoritmos de detección de comunidades multi-objetivo: Taxonomía según el tipo de representación.

Cabe señalar que el objetivo que se persigue en la optimización de un problema multi-objetivo, no solo es la obtención del conjunto de Pareto y su respectiva Frontera de Pareto óptima. Además se intenta que las soluciones estén distribuidas de manera uniforme a lo largo del PF. Esta última propiedad permite en principio la no pérdida de información útil [3,34]. De esta forma, las diferentes propuestas de algoritmos multi-objetivos deben contemplar de manera general tres aspectos fundamentales: (1) la asignación de un valor de aptitud a las soluciones óptimas de Pareto, (2) la preservación de diversidad en el conjunto de Pareto y (3) la utilización de las mejores soluciones encontradas en el proceso de búsqueda [34]. A continuación se describen dos métodos ampliamente utilizados en el contexto de la detección de comunidades siguiendo un enfoque multi-objetivo. Estos métodos diseñan y/o aplican los tres aspectos señalados anteriormente, de diversa forma.

3.2.1. Algoritmo genético basado en ordenamiento de no dominancia (NSGA-II, por sus siglas en inglés)

NSGA-II pertenece a los algoritmos evolutivos multi-objetivo que podrían denominarse tradicionales, específicamente a los basados en elitismo. En este sentido, mantiene el conjunto de mejores soluciones encontradas, soluciones élites, sin el uso de un archivo externo. De esta forma, NSGA-II mantiene en la generación t una población P_t de tamaño N y genera la próxima población P_{t+1} a partir de P_t , mediante dos pasos fundamentales [48].

1. Generar la descendencia Q_t de tamaño N a partir de P_t , mediante el uso de operadores de selección, cruzamiento y mutación.
2. Escoger las N mejores soluciones a partir de las $2N$ soluciones incluidas en $R_t = P_t \cup Q_t$ para formar la próxima población P_{t+1} de tamaño N .

Ambos pasos son desarrollados a partir del uso de un ordenamiento de las soluciones no dominadas y un concepto denominado *distancia apiñada*. De esta forma, R_t es ordenado acorde a la no dominancia, lo cual permite establecer una jerarquía u ordenamientos de frentes de Pareto. La próxima población P_{t+1} es conformada iterativamente tomando las soluciones no dominadas pertenecientes a cada jerarquía. Cuando el conjunto perteneciente a una jerarquía no puede incluirse completamente, entonces, se termina de completar las N soluciones que formarán la próxima población P_{t+1} mediante un ordenamiento utilizando la distancia apiñada [57]. Esta distancia da una medida de cuán aglomerada se encuentra una solución en el PF. Por tanto, el ordenamiento según esta métrica es en orden decreciente, de tal forma que se escogen las soluciones menos aglomeradas o apiñadas en el PF [57]. Esto último permite medir la distribución de soluciones en el PF, la cual se persigue sea lo más uniforme posible [3,34]. En la figura 4 se muestra gráficamente el método NSGA-II [57].

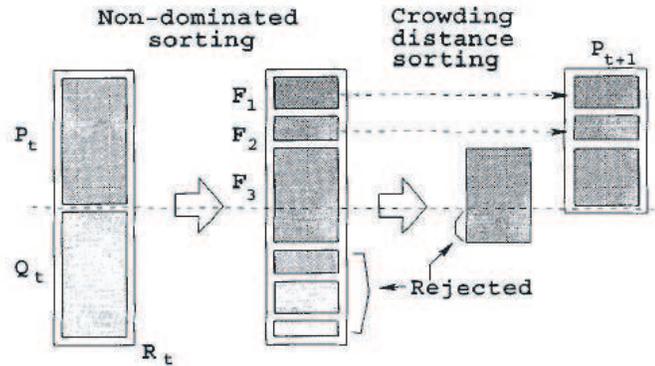


Fig. 4. Procedimiento NSGA-II [3].

Es importante destacar que la preservación del elitismo se manifiesta en la conformación del conjunto R_t , el cual incluye tanto las mejores soluciones encontradas hasta la generación t como la descendencia de esta. Luego, son ordenadas estas soluciones a partir de su dominancia. La preservación de la diversidad se ilustra una vez se tienen que escoger las soluciones finales que pertenecerán a la próxima población P_{t+1} , escogiéndose aquellas soluciones que menos aglomeradas se encuentran en el frente de Pareto [57]. NSGA-II constituye uno de los algoritmos más utilizados en el contexto de la optimización multi-objetivo.

3.2.2. Algoritmo evolutivo multi-objetivo basado en descomposición (MOEA/D, por sus siglas en inglés)

MOEA/D se basa en el enfoque tradicional de agregación. Existen diferentes enfoques de agregación como son el enfoque de sumas ponderadas y el enfoque *Tchebycheff*, entre otros [3]. Mediante un enfoque de agregación un MOPs es descompuesto en varios problemas de optimización escalares (SOPs, por sus siglas en inglés) [48]. Estos subproblemas son optimizados utilizando una población de soluciones. En cada iteración, la población es construida a partir de las mejores soluciones encontradas en cada subproblema. Cada subproblema contiene un vector de pesos asociado al conjunto de funciones objetivos, lo cual permite dar un valor de importancia a los objetivos. Estos vectores de pesos permiten establecer una relación de vecindad entre los subproblemas, a partir de la distancia existente entre ellos. Esta relación de vecindad es utilizada en el proceso de optimización para compartir las mejores soluciones encontradas entre las soluciones vecinas

[26]. Si bien MOEA/D puede utilizar cualquiera de los métodos de agregación existentes [3], el enfoque *Tchebycheff* es el utilizado por los autores del método [48].

De esta forma, el SOPs se define de la siguiente forma:

$$\min g(x|\lambda, z^*) = \max_{1 \leq i \leq q} \lambda_i |f_i(x) - z_i^*|, \text{ tal que } x \in \Omega, \quad (21)$$

donde $z^* = (z_1^*, z_2^*, \dots, z_q^*)$ constituye un punto referencia, o sea, el valor óptimo de la función i -ésima para una de las soluciones pertenecientes a Ω . La versión simple de MOEA/D archiva la mejor solución encontrada por cada subproblema a optimizar. Para cada subproblema, se genera una nueva solución utilizando operadores evolutivos que involucran la información de la vecindad del subproblema en cuestión. La mejor solución encontrada es actualizada si la nueva solución resulta más óptima. Esta solución óptima encontrada hasta el momento por el subproblema j -ésimo es compartida por los vecinos del subproblema, esto permite actualizar la mejor solución encontrada por estos subproblemas vecinos. Este principio de explotación u optimización local constituye la fortaleza del método [48,24,26].

Formalmente, sea $\lambda^1, \dots, \lambda^N$ un conjunto de vectores de pesos y z^* un punto de referencia. Mediante el uso del enfoque de agregación de *Tchebycheff*, el problema definido en la ecuación (2) se convierte en N SOPs. Luego, la función objetivo del j -ésimo subproblema [48]:

$$\min g^{te}(x|\lambda^j, z^*) = \max_{1 \leq i \leq q} \lambda_i^j |f_i(x) - z_i^*|, \quad (22)$$

donde $\lambda^j = (\lambda_1^j, \dots, \lambda_q^j)$. De esta forma, por cada solución no dominada x^* existe un vector de pesos λ tal que x^* es una solución óptima de la ecuación (22) y cada solución óptima constituye una solución no dominada del problema definido en la ecuación (2). Por tanto, MOEA/D optimiza los N SOPs en una sola corrida. Variando λ se obtiene diferentes soluciones no dominadas. Estas soluciones son mantenidas utilizando un archivo externo [48]. En este sentido, cuando se generan nuevas soluciones se verifica cuál de ellas domina alguna de las existente en el archivo externo y se reemplazan aquellas que resulten dominadas.

A manera de resumen se puede concluir que muchos de los métodos reportados del estado del arte utilizan como método de solución de problema el algoritmo NSGA-II tal y como se muestra en la figura 5.

Esta tendencia se corresponde con el hecho de que el algoritmo NSGA-II es uno de los más utilizados para abordar los problemas de optimización multi-objetivo utilizando metaheurísticas. No obstante, en los últimos años se han introducido otros enfoques de algoritmos de optimización multi-objetivo basados en metaheurísticas [24,50] que no han sido lo suficientemente explotados en el contexto de la detección de comunidades.

3.2.3. Operadores heurísticos

En los enfoques de solución utilizando metaheurísticas, los operadores heurísticos constituyen el mecanismo para construir y/o transitar por el espacio de búsqueda. Cabe destacar que estos operadores permiten modificar el estado actual del problema, a partir de su representación [58]. De esta forma, se va construyendo la vecindad del estado actual y por tanto, esta construcción define en parte los óptimos locales.

De manera general, los operadores diseñados deben permitir un correcto balance de exploración/explotación. La exploración debe permitir abarcar el mayor espacio de búsqueda del problema y de esta forma, evitar quedar atrapado en un óptimo local. En el caso de la explotación,

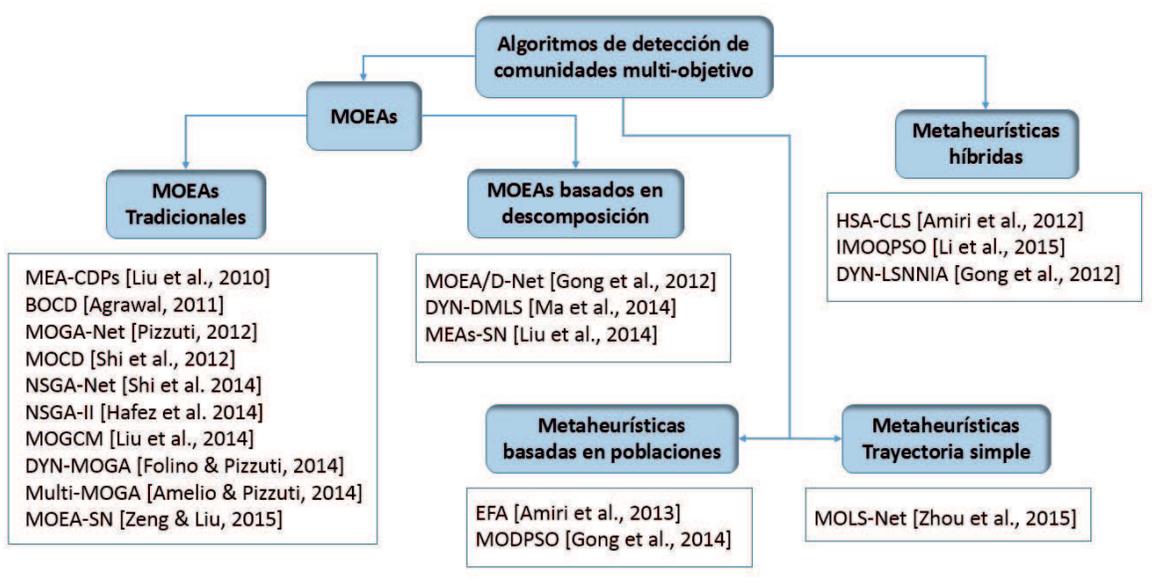


Fig. 5. Algoritmos de detección de comunidades multi-objetivo: Taxonomía según la metaheurística utilizada como método de solución de problemas.

si el espacio de estado actual es prometedor, los operadores heurísticos deben permitir profundizar la búsqueda a través de estos estados [58]. En los algoritmos evolutivos, estos operadores son denominados operadores evolutivos. Entre estos se encuentran los operadores de cruzamiento y mutación. En el caso de los operadores de cruzamiento, toman un rol protagónico en la construcción del espacio de estados del problema con mayor exploración o explotación, según su diseño. El principio de funcionamiento de los operadores de cruzamiento es intercambiar información de los padres a los descendientes y de esta forma, construir soluciones nuevas que sean más prometedoras. Este operador basa su funcionamiento, de manera general, en reglas estocásticas [58].

Por otro lado, los operadores evolutivos tienen estrecha relación con la representación del problema. En este sentido, los operadores deben generar nuevos estados que sean válidos para el problema en cuestión, a partir de la representación. Por ejemplo, en la detección de comunidades, la utilización de la representación basada en localización adyacente representa las conexiones existentes en la red. Por tanto, los operadores evolutivos utilizados sobre esta representación, deben generar un nuevo estado donde los enlaces presentes en la representación estén incluidos en la red y de esta forma constituya una solución válida del problema. En esta sección se describen algunos de los operadores evolutivos ampliamente utilizados en la detección de comunidades.

Cruzamiento uniforme

Se toman, en principio estocásticamente, dos soluciones que representan los padres a cruzar. Además, se utiliza una máscara de tamaño igual a los vectores padres, la cual constituye un vector aleatorio de 0s y 1s. Los descendientes son generados seleccionando del primer padre los genes, posiciones del vector, donde la máscara contiene ceros y del segundo padre, los genes donde la máscara tiene unos.

El uso de este tipo de operador en la detección de comunidades se debe a la posibilidad de mantener en los hijos enlaces potencialmente útiles de los padres. De hecho, a partir de una

correcta inicialización de la población, si en la posición i -ésima existe el valor j -ésimo, entonces este enlace está presente en la red. Luego del cruzamiento los hijos mantienen este enlace, debido a que estos toman en su posición i el enlace j de uno de sus padres [2]. La figura 6 ilustra un ejemplo de este tipo de operadores.

Parent1 :	5	6	6	6	2	4	8	7	8	7
Parent2 :	9	4	6	3	2	4	10	10	1	9
Mask :	1	0	1	0	0	0	1	0	1	0
Offspring	9	6	6	6	2	4	10	7	1	7

Fig. 6. Ejemplo de cruzamiento uniforme.

Cruzamiento de dos puntos

Dados dos vectores padres de dimensión N , se selecciona aleatoriamente dos puntos i y j ($1 \leq i \leq j \leq N$) y se intercambia la información entre dos padres A y B perteneciente al intervalo creado por i y j ($A_k \leftrightarrow B_k, \forall k \in \{k | i \leq k \leq j\}$). En la figura 7 se muestra un ejemplo de este tipo de cruzamiento.

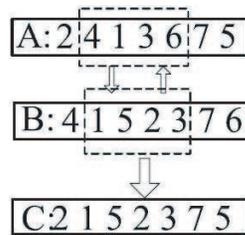


Fig. 7. Ejemplo de cruzamiento de dos puntos.

Mutación basada en vértices adyacentes

Este operador toma un individuo o cromosoma C a mutar y una posición i -ésima de este aleatoriamente, denotado como C_i . El valor de esta posición representa la posición en el cromosoma del vértice adyacente al vértice v_i . Este valor es sustituido aleatoriamente por la posición de algún vértice adyacente al vértice v_i . Es decir, la sustitución está acotada a los vecinos adyacentes del vértice v_i ($C_i \leftarrow j, j \in \{j | A_{ij} = 1\}$), siendo A_{ij} el valor de la matriz de adyacencia en las posiciones i y j . De esta forma se garantiza que los cambios se realizan solo por los nodos adyacentes, lo cual reduce el espacio de búsqueda del problema [26].

3.3. Obtención de la solución final

Como se ha detallado, en el enfoque multi-objetivo se busca identificar un conjunto de soluciones que representa un compromiso entre las funciones objetivos a optimizar. Estas funciones de manera general son contradictorias unas con otras. En este sentido, constituye un reto para el usuario decidir de estas soluciones no dominadas cuál es la solución óptima. Si bien se han desarrollado varios trabajos multi-objetivos para la detección de comunidades, poco se ha trabajado en métodos que seleccionen del conjunto de Pareto encontrado, la solución adecuada para el usuario.

La mayor parte de los trabajos reportados en el estado del arte [35,2,59,26,53,1], seleccionan la solución final del conjunto de Pareto a partir del uso de un índice de validación interno o externo.

El índice de validación interno mayormente utilizado es la Modularidad [21]. En el caso del uso de un índice de validación externo, la Información Mutua Normalizada (NMI, por sus siglas en inglés) [60] ha sido ampliamente utilizada. Esta medida se basa en la *matriz de confusión* N , donde las filas representan, en el contexto de la detección de comunidades, las comunidades reales existentes en la red, denotadas por A . Por otro lado, las columnas de N representan las comunidades identificadas por el método, denotadas por B . Los elementos pertenecientes a la matriz N , N_{ij} , representan el número de vértices de la comunidad real i que están incluidos en la comunidad identificada j . De esta forma, basado en la teoría de información, la similitud entre las dos estructuras de comunidad A y B se define como:

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{c_A} \sum_{j=1}^{c_B} N_{ij} \log \left(\frac{N_{ij} N}{N_i N_j} \right)}{\sum_{i=1}^{c_A} N_i \log \left(\frac{N_i}{N} \right) + \sum_{j=1}^{c_B} N_j \log \left(\frac{N_j}{N} \right)}, \quad (23)$$

donde c_A representa el número real de comunidades en la red y c_B el número de comunidades identificadas. N_i , N_j representan la suma de las filas i -ésimas y j -ésimas de la matriz N , respectivamente. Cabe señalar que su condición de índice externo exige contar con anterioridad con la estructura de comunidades del problema en cuestión. Esto hace su uso limitado, toda vez que en un problema no supervisado como la detección de comunidades, normalmente no se cuenta con información previa de la estructura de comunidades subyacente. De hecho, es lo que se pretende obtener con el método a desarrollar.

Por otro lado, se han desarrollado otros métodos para la selección de la solución final como la distancia mínima-máxima [4], una estrategia basada en la identificación de los puntos *Knee* y la estrategia basada en votación mayoritaria [5]. En la presente sección se caracterizan estas alternativas.

3.3.1. Distancia mínima-máxima

En [4] se proponen dos alternativas de selección de la solución final. Una de las alternativas consiste en el uso del índice Q definido en la ecuación (9). El método de distancia mínima-máxima toma el principio seguido por la Modularidad y de esta forma, MOCD o cualquier algoritmo de detección de comunidades con enfoque multi-objetivo, es ejecutado en el grafo real y en el grafo aleatorio nulo, el cual tiene la misma escala, en términos de nodos y enlaces. Luego, se identifica la solución del PF real que más desviación tiene con el PF aleatorio formado. Para la identificación de esta desviación, primeramente se define la distancia entre una solución del PF real y el PF aleatorio [4]:

$$dist(C, C') = \sqrt{(intra(C) - intra(C'))^2 + (inter(C) - inter(C'))^2}, \quad (24)$$

donde C y C' representan una solución del PF real y el PF aleatorio, respectivamente. Así mismo, *intra* e *inter* son dos funciones que miden densidad intra-grupo y dispersión inter-grupo, respectivamente, las cuales son propiedades intuitivas del concepto de comunidad [10]. La desviación de la solución del PF real con todo el PF aleatorio se calcula como la mínima distancia de esta con cualquier solución del PF aleatorio. Esta desviación, denotada como *desv*, es definida de la siguiente forma [4]:

$$desv(C, CF) = \min\{dist(C, C') | C' \in CF\}, \quad (25)$$

donde CF representa el PF aleatorio.

Por último, se selecciona del PF real aquella solución que maximiza la desviación. Por tanto, el método de selección queda formulado de la siguiente forma [4]:

$$S_{Max-Min} = \max_{C \in SF} \{dev(C, CF)\}, \quad (26)$$

donde SF represente el PF real. La figura 8 muestra un ejemplo del método de selección descrito anteriormente.

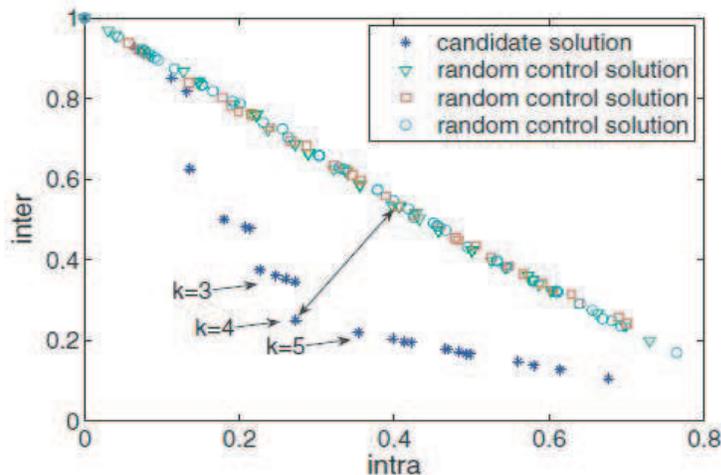


Fig. 8. Ejemplo del método de selección distancia Min-Max. La solución final es la que identifica 4 comunidades, donde k es el número de comunidades [4].

3.3.2. Estrategia basada en la identificación de los puntos *Knee*

En esta estrategia, el punto *Knee* constituye la solución que crea un punto de inflexión en la frontera de Pareto. Sea (p_1, \dots, p_n) el conjunto de soluciones no dominadas y f_1 y f_2 las funciones a optimizar, el método basado en la identificación de los puntos *Knee* consiste en los siguientes pasos [5]:

1. Identificar las dos soluciones que logran menor valor en f_1 y f_2 respectivamente;
2. Construir la línea extrema entre estas dos soluciones;
3. Calcular la distancia entre cada solución del PF y la línea extrema;
4. Identificar la solución con la máxima distancia a la línea extrema, esta es la solución *Knee*. Si existe más de una solución con valor máximo, tomar la de mayor valor en el índice Q .

La figura 9 ilustra el método descrito anteriormente.

3.3.3. Estrategia basada en votación mayoritaria

Esta estrategia utiliza la información de todas las soluciones del conjunto de Pareto. De esta forma, si dos nodos u objetos pertenecen al mismo grupo o comunidad en varias soluciones del PF entonces estos nodos se consideran pertenecientes a la misma comunidad. Sea (p_1, \dots, p_n) el conjunto de soluciones no dominadas, el método consiste en los siguientes pasos [5]:

1. Construir la matriz $A_i = (a_{kl}^{(i)})_{m \times m}$ para cada solución $p_i (1 \leq i \leq n)$, donde m es el número de nodos de la red y $a_{kl}^{(i)} = 1$ o $a_{kl}^{(i)} = 0$ si los nodos k y l pertenecen a la misma comunidad p_i o no, respectivamente.

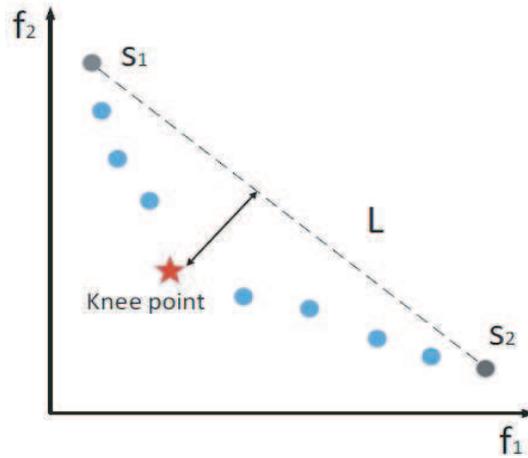


Fig. 9. Ejemplo de la identificación del punto *Knee*. S_1 y S_2 representan las soluciones de mejor valor en la función f_1 y f_2 respectivamente. L denota la línea extrema entre ambas soluciones [5].

2. Construir la matriz $B = (b_{kl})_{m \times m}$ a partir de las n matrices A_1, \dots, A_n , donde b_{kl} es calculada como:

$$b_{kl} = \frac{\sum_{i=1}^n a_{kl}^{(i)}}{n}. \quad (27)$$

3. Comparar cada elemento b_{kl} de la matriz B con un umbral dado T , reemplazando su valor a 1 si $b_{kl} \geq T$, en caso contrario se reemplaza por 0.
4. Se construye la partición basada en la matriz B : si $b_{kl} = 1$, entonces los nodos k y l pertenecen al mismo grupo o comunidad.

4. Trabajo relacionado

En la literatura se han reportado varios algoritmos de detección de comunidades a partir de un enfoque multi-objetivo. Estas propuestas han estado enfocadas en abordar diferentes tipos de redes. El tipo de red más abordada lo constituyen las redes estáticas [61,35,2,52,59,1,26,53], que son aquellas que no cambian en el tiempo. En menor medida se han abordado métodos que procesan cambios en las redes [31,32,33].

Por otro lado, los enlaces que describen la red pueden estar ponderados o no y estos pesos pueden ser positivos y negativos. Este último tipo de redes se denominan *Redes con signo* (SNs, por sus siglas en inglés) y brinda mayor información descriptiva de la red [27,62,28]. Así mismo, los vértices pudieran a su vez tener múltiples enlaces con otro vértice, representando las diferentes relaciones que pueden existir entre dos individuos o entidades. En este último caso, se dice que la red es *multi-característica*. Este tipo de red ha sido poco abordado en la detección de comunidades desde un enfoque multi-objetivo [29], tal y como se muestra en la figura 10.

En la siguiente sección se describen todos los métodos que abordan el problema de optimización multi-objetivo desde un enfoque de dominancia de Pareto, resaltando las limitaciones de cada método. En este sentido, se caracterizan sus aspectos fundamentales de diseño, los operadores heurísticos definidos, la representación utilizada, la combinación de funciones objetivos, así como la identificación de la solución final.

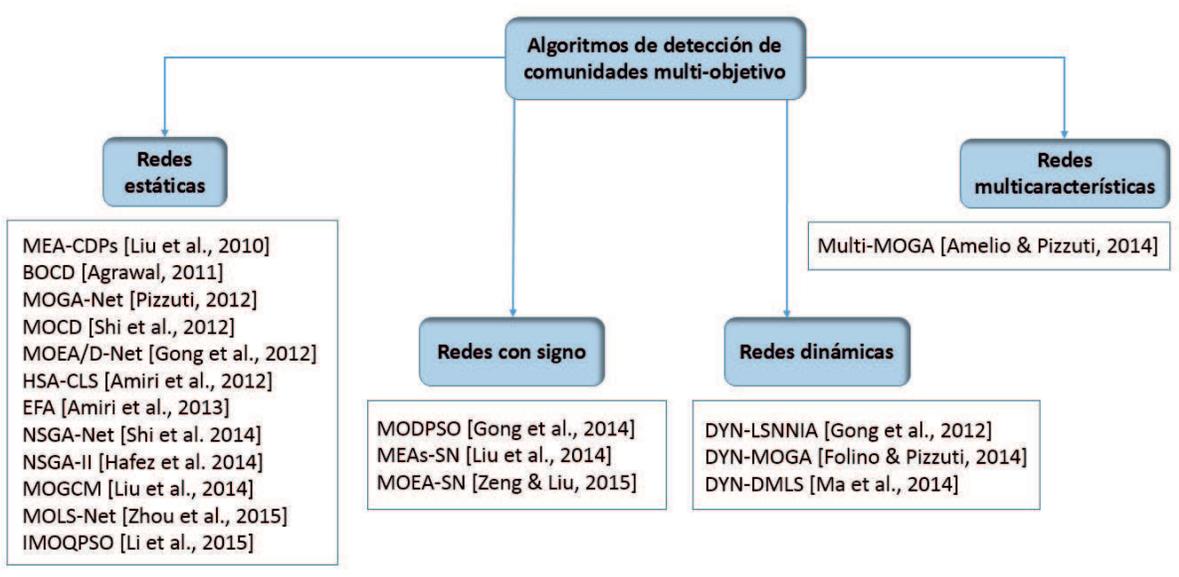


Fig. 10. Algoritmos de detección de comunidades multi-objetivo: Taxonomía según el tipo de redes que se aborda.

4.1. Algoritmos de detección de comunidades en redes estáticas

4.1.1. MEA-CDPs

El algoritmo MEA-CDPs (Multiobjective evolutionary algorithm to solve CDPs) [61], busca identificar estructuras de comunidades *duras* y solapadas. Para ello, diseña una codificación o representación de la solución indirecta basada en permutaciones, es decir, no representa explícitamente los grupos formados. Por tanto, se propone un método de construcción de los grupos. Esta representación describe un individuo A con dos componentes. La primer componente constituye una permutación de todos los nodos $v \in V$ del grafo, denotado como $\langle A \rangle = \{v_{\pi_1}, v_{\pi_2}, \dots, v_{\pi_n}\}$, donde $(\pi_1, \pi_2, \dots, \pi_n)$ es una permutación de $(1, 2, \dots, n)$. Esta representación tiene una dimensión igual al número de vértices presentes en la red, así como agrega un costo computacional en la construcción de los grupos a partir de ella.

La segunda componente de la representación constituye en sí los grupos construidos a partir de la permutación expresada por la primera componente. Esta construcción se realiza a partir de la asignación de los objetos a diferentes grupos utilizando una medida de aptitud basada en la función CF (ver ecuación (17)). El principio que se sigue es asignar el objeto al grupo si esto conlleva a un mejor valor de aptitud de la comunidad [61]. Por tanto, en esta asignación los objetos podrían ser asignados a más de una comunidad y de esta forma contemplar el solapamiento entre las comunidades.

Para construir esta segunda representación se recorre cada objeto y se calcula el valor de la función CF de cada grupo existente, en caso de adicionar a dicho objeto. Si el valor de CF del grupo aumenta entonces el objeto se adiciona a dicho grupo. En caso de que se desee formar grupos con solapamiento, el objeto puede ser adicionado a todos los grupos que satisfagan la condición anterior; en otro caso, solo se adiciona al primero que la cumpla. Si el objeto finalmente no fue adicionado a ningún grupo, entonces se crea un nuevo grupo con él. Una vez conformados los grupos, se mezclan aquellos que posean una intersección alta para mejorar las estructuras solapadas identificadas.

Se definen tres funciones objetivos, $f_{quality}$, $f_{separated}$ y $f_{overlapping}$ para guiar el proceso de búsqueda, las cuales siguen la definición dada por [10], aunque no se describe la correlación existente entre ellas. La función $f_{quality}$ mide la calidad de la estructura de comunidades identificada, la función $f_{separated}$ mide cuán separadas son estas comunidades y la función $f_{overlapping}$ el grado de solapamiento entre las comunidades. Luego, a partir de la representación, las funciones objetivos definidas y una población inicial conformada aleatoriamente, se utiliza el marco de trabajo definido por el algoritmo NSGA-II [57], descrito en la sección 3.2.1, para encontrar el conjunto de soluciones del frente de Pareto. Este método no brinda un mecanismo específico, mas allá del definido por NSGA-II, para identificar soluciones convergentes y diversas en el PF. Así mismo, no brinda una estrategia para seleccionar la solución final a partir del frente de Pareto identificado.

4.1.2. BOCD

El algoritmo BOCD (Bi-Objective Community Detection) [35], utiliza la representación basada en localización adyacente [56] descrita en la sección 3.1.2. Esta representación tiene la ventaja de permitir describir las relaciones correspondientes a la red analizada, aunque presenta la característica de ser de tamaño igual al número de vértices del grafo. BOCD busca identificar comunidades que cumplan con el concepto de comunidad dado por [10]. Para ello optimiza las funciones Q y CS, descritas en la ecuación (9) y (19), respectivamente. Cabe señalar que en el trabajo de [17] estas funciones están correlacionadas positivamente, lo cual limita la obtención de estructuras eficaces.

El método BOCD utiliza el método NSGA-II [57] para realizar el proceso de búsqueda, determinándose de manera automática el número de comunidades. Es válido mencionar que BOCD no implementa un mecanismo específico para identificar soluciones convergentes y diversas en el PF, más allá del propuesto por NSGA-II. Como operadores heurísticos, BOCD utiliza el cruzamiento uniforme descrito en la sección 3.2.3 y la mutación simple. Este último operador elige aleatoriamente un gen o posición del vector y cambia su valor. La inicialización de la población se hace de manera aleatoria, lo cual perjudica la construcción de soluciones iniciales en correspondencia a los enlaces existentes en la red. Así mismo, el operador de mutación no hace uso eficazmente de la estructura topológica de la red, debido a que realiza los cambios aleatoriamente sin contemplar los enlaces existentes. Una vez identificado el conjunto de Pareto, la solución final es aquella que mejor valor de Modularidad presente.

Cabe destacar que la Modularidad es también una de las funciones a optimizar, lo cual crea una parcialidad o sesgo en la decisión tomada para la selección de la solución final. Además, BOCD identifica estructuras de comunidades que representan particiones duras. En la experimentación se utiliza como índice de validación externo NMI y se compara respecto a los algoritmos Newman [63] y MOGA-Net [64].

4.1.3. MOGA-Net

El algoritmo MOGA-Net (Multiobjective genetic algorithms for networks) [64,2], determina de manera automática el número de grupos a partir del uso de la representación basada en localización adyacente. MOGA-Net utiliza la definición dada por [10] y utiliza el marco de trabajo definido en NSGA-II para optimizar las funciones objetivos CF y CS, descritas en las ecuaciones (17) y (19), respectivamente, fundamentándose la correlación existente entre estas dos funciones.

El método realiza una inicialización aleatoria, sin embargo, si el enlace establecido aleatoriamente para el vértice v_i no existe en el grafo, se sustituye por algún enlace existente entre el vértice

v_i y alguno de sus vecinos adyacentes. De esta forma, se hace uso de las conexiones existentes en la red de manera eficaz. El procedimiento anterior permite crear un conjunto de soluciones iniciales más acorde a los enlaces realmente existentes en la red y de esta forma, la creación de un espacio de estados iniciales más prometedores [2]. Cabe señalar que este mecanismo de inicialización es posteriormente utilizado por muchos de los trabajos reportados. MOGA-Net utiliza un operador de cruzamiento uniforme para mantener conexiones eficaces en las soluciones descendientes. Así mismo, el operador de mutación utilizado busca mantener las conexiones establecidas, para lo cual una vez escogido aleatoriamente la posición del vector a modificar, se escoge uno de los enlaces con los vértices vecinos. Tanto el operador de mutación como el operador de cruzamiento se describen en la sección 3.2.3.

De esta forma, el algoritmo comienza con la creación de la población inicial mediante una inicialización aleatoria acotada a los vértices adyacentes y luego, mientras no se cumpla una condición de terminación, se decodifican los individuos de la población y se conforman los agrupamientos resultantes de determinar las componentes conexas del grafo. Esto ayuda a la determinación automática del número de grupos o comunidades. Posteriormente, se evalúan estos agrupamientos o estructuras de comunidad en las funciones objetivos y se determina su orden según su dominancia, para luego aplicar los operadores evolutivos y conformar la siguiente generación. Esta nueva generación, junto a los padres (generación anterior), conforman el conjunto de soluciones sobre la cual se construye la próxima población. Para construir esta, se toman aquellas soluciones que mejor orden de dominancia presentan. En el caso de existir dos soluciones con igual orden de dominancia se selecciona aquella que mejor distancia apiñada presente; lo cual permite la identificación de un Frente de Pareto uniforme.

Para la selección de la solución final se escoge del conjunto de Pareto aquella que tenga mejor valor del índice de Modularidad. Así mismo, para la evaluación de la calidad en las experimentaciones se utiliza como métrica de evaluación la Modularidad y como índice de validación externo NMI. MOGA-Net se compara con algoritmos mono-objetivo ampliamente utilizados como son CNM [65], PL [66] y BGLL [67], así como con algoritmos del contexto del agrupamiento multi-objetivo como MOCK [46] y GraSC [68]. Esto ilustra la similitud que existe entre la detección de comunidades y el agrupamiento. Para realizar la comparación entre MOGA-Net y los algoritmos de enfoque multi-objetivo GraSC y MOCK se referencia el trabajo de [69] donde se plantea que un algoritmo multi-objetivo debe dar como resultado soluciones que cumplan tres propiedades: (1) minimizar la distancia de la frontera de Pareto real con la frontera de Pareto identificada, (2) que las soluciones identificadas estén distribuidas de manera uniforme sobre el espacio objetivo y (3) maximizar el número de soluciones pertenecientes al conjunto de Pareto.

En este sentido, muchas de las métricas de evaluación en el contexto de los MOPs asumen el conocimiento previo de la frontera de Pareto óptima, lo cual resulta complejo en muchos problemas prácticos. En este sentido, para comparar MOGA-Net con los algoritmos GraSC y MOCK se utiliza el índice de validación definido en [70] y denominado *espacio mínimo* (MS, por sus siglas en inglés). Por último, MOGA-Net utiliza para su validación bases de datos de gran tamaño y se compara con los algoritmos de heurística específica Newman [63] y PDB [71]. Las estructuras identificadas por MOGA-Net representan particiones duras y cabe señalar que es uno de los algoritmos de detección de comunidades con enfoque multi-objetivo más utilizado en las comparaciones realizadas por otros métodos de detección de comunidades que siguen este enfoque [4,52,59,17].

4.1.4. MOCD

El algoritmo MOCD (Multi-objective community detection) [4], a diferencia de los métodos descritos anteriormente, utiliza como método de solución el método PESA-II [72]. Esta metaheurística sigue el principio de escoger con mayor probabilidad las soluciones aisladas o menos aglomeradas. Para ellos, utiliza como unidad básica de selección el concepto de *rejilla*, la cual contiene más de una solución. El procedimiento consiste en seleccionar la *rejilla* que aglomere un gran número de soluciones, luego se escoge una solución de esta mediante un operador de selección tradicional.

MOCD descompone la función Modularidad en dos términos, $intra(C)$ e $inter(C)$, siendo C una comunidad. Estos términos están descritos en la ecuación (10) y (11), respectivamente y son utilizados como funciones a optimizar. En el caso de $intra(C)$ mide la densidad intra-grupo e $inter(C)$ mide separabilidad inter-grupo. Además, se fundamenta cómo estas funciones están correlacionadas negativamente según el número de grupos. Esta propiedad resulta de utilidad, tal y como se plantea en [4], para identificar estructuras con diferentes números de comunidades.

Por otro lado, el método utiliza la representación basada en localización adyacente, describiéndose entre sus ventajas [4]: (1) la capacidad de identificar automáticamente el número de comunidades, (2) su baja sensibilidad a generar soluciones inválidas al aplicar los operadores de cruzamiento y mutación, (3) así como la construcción de un espacio de búsqueda más reducido que el espacio generado por la representación basada en el centroide del grupo [13]. Esta representación basada en el centro del grupo resulta muy similar a la representación basada en etiqueta, descrita en la sección 3.1.1 y define un espacio de solución de complejidad espacial n^n . Por otro lado, la representación basada en localización adyacente define un espacio solución de complejidad espacial $\prod_{i=1}^n d_i$, siendo n el número de vértices y d_i el grado del vértice i . En la práctica, normalmente $d_i \ll n$ [4].

MOCD utiliza el operador de cruzamiento de dos puntos y el operador de mutación basado en los vértices adyacentes. Además, propone dos métodos para la selección de la solución final. Uno basado en el índice Q y otro que utiliza la distancia entre el frente Pareto real y varios frentes de Pareto generados aleatoriamente. A partir de las dos propuestas de selección de la solución final, se constituyen dos métodos, MOCD-Q y MOCD-D. En el caso de MOCD-Q utiliza el método de selección basado en el índice de Modularidad. Por su parte, MOCD-D hace uso del método min-max descrito en la sección 3.3.1.

En la experimentación se compara con el algoritmo ampliamente utilizado de Newman and Girvan, GN [11], el algoritmo basado en la teoría de la información denominado INFO [20] y un algoritmo mono-objetivo basado en algoritmo genético, GACD [15]. Así mismo, se comparan las alternativas MOCD-Q y MOCD-D con MOGA-Net, el cual utiliza el método min-max para la selección de la solución final. Como métrica de evaluación se utiliza el índice de validación externo relación de los vértices correctamente identificados (FVIC, por sus siglas en inglés) utilizados en otros trabajos [12,11]. FVIC es calculado de la siguiente forma:

$$FVIC = \sum_{c \in C_F} \frac{maxOlSet(c, C_K)}{N}, \quad (28)$$

donde $maxOlSet(c, C_K) = \max_{c' \in C_K} \{|olSet(c, c')|\}$ y $olSet(c, c') = \{v | v \in c \wedge v \in c'\}$. C_F y C_K representan la estructura de comunidad identificada y la estructura de comunidad real, así como c y c' son las comunidades pertenecientes a C_F y C_K , respectivamente. Por su parte, N representa el número de vértices de la red. A mayor valor de FVIC mejor estructura de comunidad identificada. Este índice al igual que NMI, asumen el conocimiento con anterioridad de las particiones, lo cual resulta poco práctico en problemas reales.

Como parte del análisis del método MOCD, se estudia la correlación existente entre la Modularidad y el número de grupos y cómo el método MOCD tiende a dar resultados con mayor granularidad que las alternativas mono-objetivo [4]. Si bien MOCD identifica estructuras de comunidades con diferente granularidad respecto al número de grupos, no contempla pesos en las relaciones entre los vértices. Por otro lado, si bien en la experimentación se muestra cómo el método identifica estructuras de comunidad con solapamiento, no se evidencia el procedimiento mediante el cual estas estructuras son determinadas.

4.1.5. MOEA/D-Net

El algoritmo MOEA/D-Net (Community detection algorithm based on a multiobjective evolutionary algorithm with decomposition) [26], constituye la primera propuesta en utilizar un MOEA basado en descomposición [48]. De esta forma, MOEA/D-Net descompone el MOPs en varios SOPs mediante el uso del método de agregación *Tchebycheff* definido en la ecuación (21). El método de agregación contempla las funciones objetivos NRA y RC descritas en la ecuación (14) y (15), respectivamente. Estas funciones están correlacionadas negativamente en función del número de comunidades, lo cual permite identificar estructuras con diferentes números de comunidades [26]. Se utiliza la representación basada en localización adyacente así como una inicialización aleatoria, restringida a los enlaces con los vértices adyacentes.

Los operadores heurísticos utilizados son el cruzamiento de dos puntos y el operador de mutación aleatorio basado en los vértices adyacentes. El hecho de acotar tanto la inicialización como la mutación solo a los vértices adyacentes, permite definir un espacio de estados más eficaz, debido a que se contemplan solo aquellos estados que incluyen enlaces contenidos en la red [26]. La obtención de la solución final se logra mediante el uso del índice Q. Durante la experimentación, MOEA/D-Net se compara con tres algoritmos mono-objetivo. El primero de ellos es el ampliamente utilizado algoritmo de Newman [63], se realizan comparaciones con el algoritmo Meme-Net [73] basado en una metaheurística meméntica que combina un algoritmo genético con un escalador de colina. Por último, se utiliza el método InfoMap [74] para comparar las alternativas de búsqueda local que brinda este respecto a MOEA/D-Net. No se realiza comparación con ningún método que utilice un enfoque multi-objetivo.

En la evaluación de la calidad se utilizan como métricas el índice de validación interno Q y el índice de validación externo NMI. No se define un mecanismo específico para identificar soluciones convergentes y diversas en el PF, mas allá del definido por MOEA/D. Además, solo contempla la identificación de comunidades disjuntas.

4.1.6. HSA-CLS

El algoritmo HSA-CLS (Multi-objective harmony search algorithm) [52], utiliza una metaheurística híbrida compuesta por un algoritmo de búsqueda armónica (HSA, por sus siglas en inglés) [75] y un algoritmo de búsqueda local caótico (CLS, por sus siglas en inglés) [76]. La búsqueda local caótica es utilizada para atenuar la convergencia rápida hacia un óptimo local del método de búsqueda armónica [52]. En este sentido, luego de la inicialización se aplica la búsqueda local caótica sobre las soluciones contenidas en la Memoria Armónica (HM, por sus siglas en inglés) obteniéndose soluciones en principio mejoradas a partir de CLS. Estas nuevas soluciones obtenidas a partir de CLS son evaluadas en las funciones objetivos y aquellas que resulten no dominadas son comparadas con las soluciones no dominadas incluidas en la HM. A partir de esta operación

se actualiza la HM. Por tanto, HSA-CLS utiliza HM como un repositorio externo para archivar las soluciones no dominadas encontradas durante el proceso de búsqueda.

Cabe señalar que la construcción del repositorio HM a partir de las soluciones obtenidas mediante la aplicación de la búsqueda local CLS, se realiza como parte de la inicialización de la población. A partir de esto, se realiza un proceso iterativo de mejoras de soluciones, aplicando los operadores definidos por el algoritmo de búsqueda armónica y posteriormente, se actualiza el repositorio reemplazando aquellas soluciones que resulten dominadas por las nuevas soluciones que se van creando. Por otro lado, HSA-CLS utiliza la representación basada en localización adyacente, lo cual permite la identificación automática del número de comunidades.

HSA-CLS optimiza las funciones CF y CS, comentadas en la descripción del algoritmo MOGA-Net (ver sección 4.1.3). HSA-CLS no define un método de selección de la solución final, aunque utiliza como medida de evaluación Q y NMI en los experimentos, donde se compara contra el algoritmo de heurística específica CNM [65] y el algoritmo multi-objetivo MOGA-Net [2]. No se describe el tipo de inicialización utilizado, así como no se brinda un mecanismo específico para identificar soluciones convergentes y diversas en el PF. Las soluciones obtenidas por HSA-CLS representan particiones duras.

4.1.7. EFA

El algoritmo EFA (Enhanced Firefly Algorithm) [59], utiliza una metaheurística poblacional basada en SI, denominada *Algoritmo de las luciérnagas* (FA, por sus siglas en inglés) [77]. Este algoritmo tiene la característica de utilizar varios parámetros, tales como el *coeficiente de atracción* y el *factor de movimiento aleatorio*, los cuales, si son mal calibrados, pueden provocar que el algoritmo quede atrapado en un óptimo local. En tal sentido, EFA propone un mecanismo automático de calibración de los parámetros basado en secuencia caótica [78], el cual es utilizado también para la obtención de la población inicial. Por otro lado, EFA utiliza la representación basada en localización adyacente y por tanto, identifica automáticamente el número de comunidades. Las funciones objetivos que guían el proceso de búsqueda son CF y CS, al igual que los algoritmos MOGA-Net y HSA-CLS, descritos en la sección anterior.

EFA utiliza un repositorio externo para almacenar las soluciones no dominadas encontradas; las soluciones no dominadas pertenecientes al repositorio que resulten dominadas por soluciones nuevas son reemplazadas. Para cumplir con una de las propiedades deseables en la identificación de la frontera de Pareto, se propone un mecanismo de preservación de diversidad basado en nicho. En este sentido, se busca identificar la mejor solución (denotada como $Best^k$) del conjunto de soluciones no dominadas para apoyar que las soluciones no queden distribuidas en una misma región.

La identificación de $Best^k$ se realiza a partir del uso de una función de distancia compartida $d_{n_1 n_2}$ que mide qué tan diferentes son las evaluaciones de dos soluciones n_1 y n_2 en las funciones objetivos a optimizar. Así mismo, se define una función que calcula el recuento de nichos a partir del uso de $d_{n_1 n_2}$, de tal forma que este recuento resulta pequeño si existe mucha distancia compartida entre n_1 y n_2 . A partir de lo anterior, se define un procedimiento de asignación de aptitud a las soluciones pertenecientes al conjunto de Pareto, en donde aquellas soluciones que presenten valores en las funciones objetivos muy distantes del resto tendrán un valor de aptitud alto. Esto indica que la solución presenta poco nicho a su alrededor, es decir, se encuentra poco aglomerada. Por tanto, la aptitud asignada a ella constituye el inverso del recuento de nichos. Luego, mediante el uso del operador de selección de la ruleta [79] se selecciona $Best^k$.

Por otro lado, para mantener controlado el tamaño del repositorio, se introduce un procedimiento que calcula un grado de pertenencia difuso de las soluciones pertenecientes al frente, teniendo en cuenta la evaluación de estas en las funciones objetivos. Además, este procedimiento utiliza una asignación de pesos dada a las funciones objetivo por medio del experto del dominio de aplicación. Mediante estos pesos, junto con el grado de pertenencia calculado, se conforma un índice denominado grado de pertenencia normalizado y pesado, el cual permite guiar la selección de las soluciones que se mantendrán en el repositorio, hacia aquellas que mejor cumplen los requerimientos o preferencias del experto [59]. Estas preferencias son expresadas a través de las ponderaciones dadas. Para apoyar mejor el proceso de búsqueda, se define un operador de mutación probabilístico adaptativo que permite mejorar la calidad de las soluciones y controlar la convergencia a un óptimo local [59]. Como insuficiencias, EFA no contemplan pesos en los enlaces presentes en la red y construye estructuras que representan particiones duras.

4.1.8. *NSGA-Net*

Tal y como se comentó en la sección 2.3.3, las funciones objetivos que se pretenden optimizar deben en principio cumplir estar relacionadas negativamente; es decir, se comporten de manera opuesta respecto a algún parámetro, como por ejemplo el número de comunidades. Este resultado es fundamentado en el trabajo de [17] mediante el estudio de varias de las funciones objetivas utilizadas en la detección de comunidades. En el trabajo, se pretende dar respuesta a preguntas tales como: ¿Cómo afecta la optimización de varias funciones objetivos el rendimiento en la detección de comunidades multi-objetivo?, ¿Qué tipo de funciones objetivos deben ser optimizadas para lograr una mejor eficacia en las comunidades identificadas?

Por tanto, el método NSGA-Net no constituye un nuevo algoritmo sino más bien el marco de trabajo sobre el cual se experimentan diferentes funciones objetivos, correlacionadas de diversas formas, para medir la eficacia de los resultados obtenidos. Los diferentes pares de funciones objetivos analizadas fueron agrupadas según su correlación en: (1) funciones correlacionadas positivamente, (2) funciones correlacionadas negativamente y (3) funciones no correlacionadas. A partir de esto y utilizando NSGA-II como método de solución de problema, se experimentaron 6 pares de funciones objetivos pertenecientes a las clasificaciones mencionadas, dos por cada clase. Se compararon los resultados obtenidos por NSGA-Net contra un método de optimización mono-objetivo basado en algoritmo genético.

Si bien NSGA-Net no representa en si un nuevo algoritmo y constituye más bien un marco de trabajo, sobre este se definen aspectos que se han descrito en las secciones previas. En este sentido, se utiliza la representación basada en localización adyacente así como una inicialización aleatoria, aunque esta última acotada a los vértices adyacentes. Esta acotación se realiza con el objetivo de mantener soluciones válidas en el proceso de búsqueda. Con este mismo propósito, se utiliza el operador de cruzamiento de dos puntos y el operador de mutación basado en los vértices adyacentes. Se propone como método de selección de la solución final el método descrito en la sección 3.3.1.

Los diferentes experimentos realizados permiten un mejor análisis del tipo de estructura correspondiente a las diferentes funciones objetivos estudiadas. Así mismo, se demostró como aquellas funciones objetivos correlacionadas negativamente son las que mejores índices de eficacia presentan [17]. No obstante, varias de las funciones objetivos descritas en la sección 2.3 y que son ampliamente utilizadas por los métodos propuestos del estado del arte no son tenidas en cuenta. Así mismo, todos los experimentos están orientados a detectar comunidades disjuntas.

4.1.9. *GAsMCD*

Esta propuesta [80] se basa en el algoritmo NSGA-II y al igual que el trabajo comentado en la sección previa, define un marco de trabajo sobre el cual se analizan diferentes funciones objetivos, analizadas en los trabajos de [18] y [47]. El trabajo persigue un objetivo similar al trabajo descrito en [17], donde se optimizan diferentes pares de funciones objetivos buscando identificar aquellas que mejores índices de eficacia brindan. Los experimentos realizados utilizan primeramente un algoritmo genético mono-objetivo, el cual optimiza cada una de las funciones objetivos analizadas. El análisis se realiza tanto en bases de datos reales como sintéticas promediando el índice de validación externo NMI tras varias corridas del algoritmo por cada función objetivo. Con este análisis se busca identificar cuál de las funciones objetivos resulta con mejor valor NMI para las bases de datos utilizadas.

Así mismo, se realiza un análisis para identificar cuáles de las funciones objetivos resultan más estables. Definen como función objetivo estable aquella que es capaz de asignar valores de aptitud similares ante estados del problema que son similares. Debido a que la función objetivo es en gran medida la que guía el proceso de búsqueda, ante funciones objetivos no estables, los algoritmos darán resultados diferentes sobre el mismo conjunto de datos, lo cual no resulta deseable en la práctica debido a la poca consistencia en los resultados obtenidos. Por tanto, en el trabajo definen como función objetivo estable aquella que es capaz de brindar estructuras de comunidad similares ante diferentes corridas del algoritmo sobre el mismo conjunto de datos [80].

En el caso del algoritmo multi-objetivo, se experimenta con todos los pares de funciones objetivos para identificar cuál de ellas brinda mejor valor de eficacia mediante el cálculo del índice de validación externo NMI. Este algoritmo basado en NSGA-II, utiliza una representación basada en localización adyacente. Además, al igual que el método NSGA-Net descrito en la sección previa, implementa una inicialización aleatoria pero acotada a los vértices adyacentes, con el objetivo de mantener soluciones válidas. Con el mismo objetivo de mantener el proceso de búsqueda sobre un espacio de estados válidos y en mayor correspondencia con la red real, se utiliza el operador de cruzamiento uniforme y la mutación basada en los vértices adyacentes.

La propuesta, al igual que la anterior NSGA-Net, está orientada a redes disjuntas, además de no contemplar varias funciones objetivos ampliamente utilizadas en la detección de comunidades con un enfoque multi-objetivo. Cabe agregar que ambas propuestas se concentran en un análisis acerca de las funciones objetivos, dejando a un lado el diseño de alternativas que busquen con intencionalidad converger a una Frontera de Pareto óptima y uniforme.

4.1.10. *MOGCM*

El algoritmo MOGCM (Multiobjective genetic community discovery method) [62], optimiza las funciones objetivos MMC y CF descritas en la ecuación (20) y (17), respectivamente. Para ello, utiliza la representación basada en localización adyacente así como una inicialización basada en una lista ordenada de sus vecinos adyacentes. Con esta inicialización, se persigue obtener estados iniciales que describan mejor los enlaces existentes entre los vértices de la red.

Para ir creando el espacio de estados del problema, MOGCM utiliza como operadores heurísticos el operador de cruzamiento de dos puntos y el operador de mutación acotado a los vértices adyacentes. Ambos operadores están descritos en la sección 3.2.3. En el operador de mutación, al igual que la construcción de la población inicial, se hace uso de los vértices adyacentes del vértice a mutar y por tanto, de la información topológica de la red.

Procedimentalmente se construye el conjunto de soluciones iterativamente, utilizando la metaheurística GAs y estableciendo un ordenamiento o jerarquía similar al que emplea el algoritmo

NSGA-II. Para generar nuevas soluciones se toman dos soluciones pertenecientes a la mayor jerarquía de no dominancia y se aplican los operadores de cruzamiento y mutación. Como solución final se devuelve aquella que mejor índice Q presente. Las métricas de calidad utilizadas en las experimentaciones son los índices Q y NMI. El algoritmo MOGCM se compara solamente con el algoritmo de heurística específica GN [11]. Por otro lado, se realiza una experimentación reemplazando la función objetivo CF por CS, evidenciando mejores resultados obtenidos a partir del índice NMI. En ninguna de las dos combinaciones de funciones objetivos utilizadas, MMC y CF o MMC y CS, se hace un análisis de su correlación.

Como insuficiencias, no se contempla pesos en los enlaces presentes en la red, así como se determinan estructuras que representan particiones duras. Por último, MOGCM no implementa un mecanismo específico para identificar soluciones convergentes y diversas en el PF.

4.1.11. *MOLS-Net*

El algoritmo MOLS-Net (Local search-based multiobjective optimization algorithm) [1], es la primera propuesta de algoritmo multi-objetivo para el CDPs que utiliza metaheurísticas de búsqueda local. En este sentido, dada una solución X y dos funciones objetivos a optimizar, MOLS-Net desarrolla dos búsquedas locales, $LS_1(X)$ y $LS_2(X)$ para optimizar la función f_1 y f_2 , respectivamente. El método utiliza un archivo de soluciones no dominadas que se actualiza a medida que se construyen nuevas soluciones. El procedimiento general de MOLS-Net consiste en: (1) seleccionar una solución perteneciente al archivo, (2) aplicar una búsqueda local sobre esta solución por cada función objetivo y (3) actualizar el archivo de soluciones no dominadas.

Por otro lado, MOLS-Net utiliza la representación basada en etiqueta descrita en la sección 3.1.1 y construye una población inicial mediante el procedimiento descrito en [81]. Este procedimiento consiste en marcar inicialmente cada vértice con una etiqueta única. Luego, se establece un orden secuencial aleatorio entre todos los vértices y se recorre esta secuencia asignando a cada vértice la etiqueta que poseen la mayoría de sus vecinos. Este recorrido se realiza iterativamente hasta lograr una convergencia en donde cada vértice tiene la etiqueta de la mayoría de sus vecinos. En este punto, cada grupo de vértices con etiquetas idénticas se considera una comunidad [1]. Esto permite crear estados iniciales más prometedores para el proceso de búsqueda, pero aumenta el costo computacional.

Las funciones objetivos que optimiza MOLS-Net son RC y *Kernel K-Means* descritas en la ecuación (16) y (15), respectivamente. Válido señalar que la ecuación (15) describe la función denominada MRA [41]. No obstante, tal y como se describe en [27], esta función es denominada también Kernel K-Means (KKM, por sus siglas en inglés). Las funciones $f_1 = RC$ y $f_2 = KKM$ son minimizadas mediante los métodos de búsqueda local LS_1 y LS_2 , a partir de operaciones de mezclar y dividir comunidades, respectivamente. En el caso del método LS_1 , la operación de mezcla decrementa el número de comunidades a la vez que se minimiza la función RC. Por otro lado, el método LS_2 divide las comunidades incrementando el número de estas y minimizando la función KKM. Esta estrategia de división está basada en el método BGLL [67] y es la que introduce mayor costo computacional [1].

Durante la optimización, dos situaciones se pueden presentar: (1) las nuevas soluciones dominan a X y por tanto, se logra una mejor convergencia de la frontera de Pareto y (2) las nuevas soluciones generadas no dominan ni son dominadas por X , por tanto se logra mayor diversidad sobre la frontera de Pareto. Estos casos representan convergencia y diversidad, respectivamente. Las soluciones deseables son aquellas con mejor convergencia a la frontera de Pareto y distribuidas uniformemente a lo largo de esta [3,34].

Es válido destacar, que los conceptos de convergencia y uniformidad no se encuentran rigurosamente definidos en los métodos reportados. Un análisis en este sentido, apoyaría metodológicamente el diseño de nuevos algoritmos. MOLS-Net archiva estas soluciones en el conjunto de soluciones no dominadas. Este archivo se actualiza durante el proceso de búsqueda para evitar perder cualquier solución no dominada. El método está orientado solo a determinar particiones duras.

En la experimentación se realiza una comparación con los algoritmos multi-objetivo MODPSO [27] y MOEA/D-Net [26]. Así como se experimenta con los algoritmos mono-objetivos GA-Net [14] y BGLL [67]. Para la evaluación de la calidad de los métodos analizados se utilizan las métricas Q y NMI.

4.1.12. IMOQPSO

El algoritmo IMOQPSO (Improved multi-objective quantum-behaved particle swarm optimization) [53], es de los pocos trabajos reportados que aborda la detección de comunidades con solapamiento. En este sentido, a partir del grafo original $G = (V, E)$ se define el grafo lineal $L(G)$, en el cual el número de vértices coincide con el número de enlaces pertenecientes a G . Esta característica se debe a que el grafo lineal cumple dos propiedades: (1) cada vértice en $L(G)$ representa un enlace en G y (2) dos vértices de $L(G)$ son adyacentes si y solo si los enlaces correspondientes a ellos tienen un vértice en común. Una vez transformado el grafo original en un grafo lineal, se utiliza un método espectral para extraer la información espectral sobre el grafo transformado, así como en la inicialización de la población. Señalar que el uso del grafo lineal como mecanismo para abordar estructuras de comunidades con solapamiento no es nuevo pero si su combinación con un enfoque multi-objetivo [53].

IMOQPSO define el concepto de matriz de vecinos comunes. Esta matriz representa el número de vértices adyacentes que tienen en común dos vértices v_i y v_j . A partir de esta matriz, se redefine la función Modularidad, definiéndose la *Modularidad de vecinos comunes*. Las funciones a optimizar por IMOQPSO son los dos términos que componen la función Modularidad de vecinos comunes. IMOQPSO define una representación híbrida basada en un vector de centroides y en un vector de valores aleatorios pertenecientes al intervalo $[0 - 1]$; los valores contenidos en este último vector representan la activación o no de los centroides. Esta representación tiene el inconveniente de fijar con antelación una cantidad máxima de grupos.

El proceso de optimización se realiza mediante el uso del método de optimización basado en partícula con comportamiento cuántico (QPSO, por sus siglas en inglés) [82,83] y el algoritmo de búsqueda armónica [75]. El método QPSO genera las nuevas soluciones y la búsqueda armónica mantiene la diversidad en el enjambre de partículas construidas. Se utiliza un repositorio de tamaño fijo para mantener las soluciones no dominadas encontradas. Si el número de soluciones a incluir en el repositorio sobrepasa el tamaño fijado, se priorizan aquellas soluciones que menos aglomeradas se encuentran en el PF y de esta forma, se mantiene la diversidad. Terminado el proceso de optimización, se define un operador de calibración fina para mejorar los grupos de baja precisión. Por último, se utiliza la medida Modularidad, modificada para comunidades con solapamiento, así como la modificación de NMI para abordar solapamiento, propuesto por [43], para seleccionar la solución final.

En la experimentación se utilizan bases de datos sintéticas y reales y se compara con algoritmos de enfoque multi-objetivo y mono-objetivo. Entre las alternativas comparadas con enfoque multi-objetivo se encuentra MOPSO propuesta por [84], así como el algoritmo ampliamente utilizado

NSGA-II [57]. Por su parte, entre los métodos mono-objetivo se encuentran GA-Net+ [85] y OH [43]. GA-Net+ utiliza el concepto de grafo lineal para abordar el solapamiento.

A manera de resumen se podría decir que IMOQPSO utiliza una combinación de enfoques y técnicas para abordar el solapamiento. En este sentido, utiliza el concepto de grafo lineal para transformar la detección de comunidades solapadas en el grafo original G en la detección de comunidades separadas en el grafo lineal $L(G)$. Además se utiliza un método espectral para identificar un vector de características del grafo lineal y de esta forma, transformar el problema en un agrupamiento tradicional. Mediante el uso de la representación basada en centroides se busca disminuir la dimensión del vector solución. Luego, a partir de un proceso de optimización multi-objetivo se detecta el conjunto de soluciones no dominadas, las cuales son posteriormente mejoradas con un procedimiento de calibración.

4.1.13. Limitaciones o deficiencias presentes en los métodos descritos

Como se puede apreciar en las secciones anteriores, todos los métodos basan su funcionamiento en la definición de comunidad dada por [10], no obstante, esta definición no contempla información descriptiva, solo información estructural; es decir, solo utiliza la información topológica de la red como es el número de enlaces y vértices, la matriz de adyacencia, etc. Sin embargo, existen problemas donde pueden estar presentes diferentes informaciones descriptivas tanto de los vértices como de los enlaces.

Por otra parte, el concepto de comunidad no tiene incluido descripción de pesos asociados en sus enlaces, así como no permite medir la proporción de diferencia que puede haber entre los enlaces internos y externos para que una comunidad sea fuerte o débil. Por ejemplo, si dos comunidades tienen más enlaces internos que externos, pero una de ellas tiene mucha mayor densidad interna que la otra, según la definición, ambas serían comunidades fuertes o débiles no pudiéndose establecer diferencia entre ellas.

Las estructuras de comunidad obtenidas por la mayoría de los métodos propuestos representan particiones duras, existiendo solamente dos trabajos que permiten formar comunidades con solapamiento. No obstante, en el contexto de las redes sociales es común que los individuos o entidades puedan pertenecer a más de una comunidad, por lo que ser capaz de construir comunidades con solapamiento permitiría reflejar de forma más natural el problema en cuestión.

Por otra parte, muchas de las funciones utilizadas por los algoritmos en la optimización derivan de la función Modularidad, la cual se describió en la sección 2.3.3 tiende a mezclar comunidades pequeñas debido al problema de límite de resolución. Pocos trabajos realizan un análisis riguroso acerca de la correlación existente entre las funciones a optimizar, lo cual no permite medir con claridad las diferentes estructuras que se obtienen. Muchas de las funciones utilizadas en el proceso de búsqueda no son consecuentes con el concepto de comunidad que referencian. A excepción de la función densidad de la modularidad, CF y MMC, el resto solo mide la propiedad de alta densidad intra-grupo o la de alta separabilidad inter-grupo.

En la mayoría de los trabajos analizados, salvo MOCN, la obtención de la solución final, se realiza mediante el uso de un índice de validación interna y/o índice de validación externa. Es válido destacar que este último requiere el conocimiento a priori de la estructura de comunidades existente en los datos, por lo que este tipo de selección sería inapropiado en problemas reales, en los cuales por lo general se desconoce la estructura de comunidades que existe en la red. Por otra parte, los índices de validación interno utilizados miden alguna propiedad de la comunidad que no necesariamente va en correspondencia al tipo de propiedad medida por las funciones objetivos

utilizadas en el proceso de búsqueda. Esto provoca que las estructuras identificadas, podrían no dar valores correctos para el índice de validación utilizado.

La representación basada en localización adyacente fue la representación más utilizada por los métodos. No obstante, esta representación es costosa en memoria, por lo que en problemas reales en los que se procesen redes de gran tamaño, el método resultaría ineficiente.

Por último, es importante destacar que pocos métodos diseñan un mecanismo específico para identificar soluciones convergentes y diversas en el PF, lo cual puede provocar la obtención de un conjunto de Pareto representado por soluciones poco diversas y que representan óptimos locales.

4.2. Algoritmos de detección de comunidades de redes con signo

Los algoritmos descritos en la sección 4.1, están orientados a detectar comunidades en redes cuyos enlaces no poseen ponderaciones que describan si estas relaciones son positivas o no. A este tipo de redes se les denomina *Redes sin signo*. Cuando la red contiene enlaces positivos o negativos se denominan *Redes con signo* (SNs, por sus siglas en inglés). Este tipo de redes describen con mayor naturalidad las relaciones existentes en las redes sociales. Estas relaciones pueden ser positivas, denotando amistad, cooperación, confianza, etc, mientras los enlaces negativos denotan hostilidad, disgusto, desconfianza, represión, etc.

En las redes sin signo, las estructuras de comunidades están definidas a partir de la densidad de enlaces tanto intra-grupo como inter-grupo. En el caso de las SNs, las comunidades no solo están definidas a partir de la densidad de enlaces sino también se tienen en cuenta si estos enlaces son positivos o negativos. Intuitivamente, según plantean varios autores [86,27], dentro de la comunidad debe existir gran densidad de enlaces positivos, mientras que entre las comunidades deben existir muchos enlaces negativos. En la presente sección se describen los métodos de detección de comunidades que abordan este tipo de red.

4.2.1. MODPSO

El algoritmo MODPSO (Multiobjective Discrete Particle Swarm Optimization Based on Decomposition) [27], redefine el concepto de comunidad dado por [10] para SNs. Para ello, define el grafo con un conjunto de enlaces con valores positivos y un conjunto de enlaces con valores negativos. Formalmente, sea $G(V, PE, NE)$, donde V representa el conjunto de vértices y PE y NE los conjuntos de enlaces con valores positivos y valores negativos, respectivamente. La matriz de adyacencia A contiene valores 1, 0 y -1 para describir los enlaces positivos, la no existencia de enlaces, así como los enlaces negativos, respectivamente. De esta forma, sea $e_{v_i v_j}$ el enlace entre el vértice v_i y el vértice v_j . $A_{ij} = 1$ si $e_{v_i v_j} \in PE$. Por otro lado, $A_{ij} = -1$ si $e_{v_i v_j} \in NE$ y en última instancia $A_{ij} = 0$ si $\nexists e_{v_i v_j}$.

A partir de lo anterior, sea $S \subset G$ un subgrafo que contiene al vértice v_i . Los grados internos, tanto positivos como negativos, del vértice v_i se definen como $(k_{v_i}^+)^{in}(S) = \sum_{v_j \in S, e_{v_i v_j} \in PE} A_{v_i, v_j}$ y $(k_{v_i}^-)^{in}(S) = \sum_{v_j \in S, e_{v_i v_j} \in NE} |A_{v_i, v_j}|$, respectivamente. Así mismo, los grados externos de un vértice v_i , tanto negativos como positivos, se definen como $(k_{v_i}^-)^{out}(S) = \sum_{v_j \notin S, e_{v_i v_j} \in NE} |A_{v_i, v_j}|$ y $(k_{v_i}^+)^{out}(S) = \sum_{v_j \notin S, e_{v_i v_j} \in PE} A_{v_i, v_j}$, respectivamente. Consecuentemente, se redefinen los conceptos de comunidad fuerte y comunidad débil. De esta forma, una comunidad es fuerte si:

$$(k_{v_i}^+)^{in}(S) > (k_{v_i}^-)^{in}(S), \quad \forall v_i \in S, \quad (29)$$

o sea, si se cumple que todos los vértices dentro de ella tienen más enlaces internos positivos que negativos. Por otro lado, MODPSO define una comunidad débil como aquella que cumple:

$$\begin{cases} \sum_{v_i \in S} (k_{v_i}^+)^{in}(S) > \sum_{v_i \in S} (k_{v_i}^+)^{out}(S), \\ \sum_{v_i \in S} (k_{v_i}^-)^{out}(S) > \sum_{v_i \in S} (k_{v_i}^-)^{in}(S), \end{cases} \quad (30)$$

es decir, (1) la suma de los grados internos positivos es mayor que la suma de los grados externos positivos; o sea, existen dentro de la comunidad más enlaces positivos que fuera de esta y (2) la suma de los grados externos negativos es mayor que la suma de los grados internos negativos; es decir, los enlaces negativos son mayores con los nodos externos a la comunidad que con los nodos internos. La definición anterior mide propiedades útiles que deben cumplir las SNs, no obstante, resulta de interés medir también la densidad de enlaces negativos dentro de la comunidad y la densidad de enlaces positivos entre comunidades.

MODPSO redefine las funciones objetivos NRA y RC (ver ecuación (14) y (16), respectivamente) para SNs, denominándolas SRA y SRC, respectivamente. Estas funciones están correlacionadas negativamente respecto al número de grupos, lo cual permite identificar estructuras de comunidades más eficaces. Ambas funciones se minimizan para identificar comunidades que tengan gran densidad de enlaces positivos dentro de la comunidad, así como muchos enlaces negativos entre comunidades. MODPSO utiliza una representación basada en etiqueta y la metaheurística PSO para problemas discretos, en el proceso de optimización. Además, se utiliza la descomposición del método MOEA/D para lograr mayor diversidad en las soluciones generadas. De esta forma, una vez dividido el problema en varios subproblemas, cada partícula optimiza uno de los subproblemas acorde al vector de pesos, lo cual permite mayor distribución de soluciones a lo largo del PF, así como mitigar la convergencia prematura [27].

MODPSO realiza una inicialización de las soluciones, partículas, mediante un método de propagación de etiquetas [81]. Adicionalmente, para el proceso de búsqueda, MODPSO define también un operador de turbulencia, denominado *mutación basado en vecinos* con el objetivo de buscar mayor exploración/explotación del espacio de búsqueda. De esta forma, se comienza la búsqueda con una turbulencia alta para lograr una mayor exploración del espacio de búsqueda y luego se va disminuyendo esta turbulencia para intensificar la búsqueda por aquellas regiones más prometedoras. El método identifica estructuras de comunidad que representan particiones duras.

Se realiza una amplia experimentación sobre varios algoritmos tanto multi-objetivos como mono-objetivos. En este sentido, se realiza comparaciones con varias alternativas evolutivas como son GA-Net [14], Meme-net [73], MOGA-Net [2], MOCD [4], MOEA-D-Net [26] y dos alternativas basadas en el método MOPSO [84]. Así mismo, MODPSO utiliza un mecanismo de descomposición al igual que MOEA-D-Net y compara su mecanismo de convergencia rápida contra el mecanismo utilizado por Meme-net. Por otro lado, se aborda la comparación con los métodos de heurística específica tales como GN [11], CNM [65] e Informap [74]. Se utiliza como métricas de evaluación los índices Q y NMI.

4.2.2. MEAs-SN

El algoritmo MEAs-SN [86], propone dos funciones objetivos para medir alta densidad de enlaces positivos dentro de la comunidad y maximizar los enlaces negativos entre las comunidades. Para ello, redefine la función definida por [87] para abordar enlaces positivos y negativos, mediante el uso de la teoría de *balance social* [88]. En este trabajo se fundamenta la correlación negativa que presentan las funciones definidas, lo cual permite identificar comunidades más eficaces [17].

MEAs-SN identifica estructuras de comunidades solapadas. Para lograr este objetivo, se utiliza una representación indirecta basada en permutaciones, similar a la definida en [61]. Además, para una mayor eficiencia en la evaluación de las soluciones, se conjuga la representación anterior con una representación basada en etiqueta. En el método de construcción de los grupos a partir de la representación basada en permutaciones, se puede agregar un vértice a diferentes comunidades. Para ello se hace uso de una medida que indica si la comunidad mejora una vez agregado el vértice analizado. Además, se mezclan los grupos que tengan muchos vértices en común para mejorar las comunidades solapadas encontradas.

MEAs-SN diseña operadores de cruzamiento y mutación para cada una de las representaciones utilizadas. En este sentido, la representación basada en permutaciones utiliza un operador de cruzamiento de unificación parcial [89] y un operador de mutación que escoge aleatoriamente dos elementos de la permutación y los intercambia. Esta representación es utilizada para lograr inicialmente mejores estructuras de grupos con solapamiento. Luego, se utiliza la representación basada en etiqueta para una mayor eficiencia en la búsqueda y se utiliza el operador de cruzamiento utilizado en [13]. El operador de mutación hace uso de los vecinos adyacentes con enlaces positivos así como de la similitud con estos.

MEAs-SN utiliza el marco de trabajo definido por MOEA/D, con el enfoque de agregación descrito en la ecuación (21), además de utilizar ambas representaciones durante el proceso de búsqueda. La representación basada en permutaciones es utilizada en la inicialización y en la construcción de comunidades solapadas, la representación basada en etiqueta permite mayor eficiencia en el proceso de optimización y está orientada a la búsqueda de grupos duros. Por tanto, MEAs-SN permite identificar tanto comunidades solapadas como comunidades duras, sin el conocimiento previo del número de grupos. Por último, es importante destacar que MEAs-SN no implementa un mecanismo específico para identificar soluciones convergentes y diversas en el PF.

4.2.3. MOEA-SN

El método MOEA-SN [28], está basado en NSGA-II y redefine la función Modularidad para abordar las SNs. Para ello define dos funciones Q^+ y Q^- que representan la desviación de la estructura de comunidad en el grafo real con el grafo aleatorio nulo asociado a este, tanto para los enlaces positivos como negativos. MOEA-SN utiliza la representación basada en etiqueta, lo cual representa particiones duras. Además se utiliza el método de propagación de etiquetas definido por [81] para la inicialización de las soluciones iniciales. Este procedimiento de inicialización buscar construir estados iniciales más prometedores para el proceso de búsqueda.

MOEA-SN utiliza como operador de cruzamiento el definido en [14] y como operador de mutación el descrito en [90] el cual se basa en búsqueda local. Esta búsqueda local como heurística específica toma el fundamento definido por la función de Modularidad. Señalar que la Modularidad es redefinida para el contexto de las redes con signo. De esta forma, una vez seleccionado el vértice a mutar v_i , se realiza una búsqueda por aquellos nodos adyacentes que no posean la misma etiqueta de grupo y se modifica la etiqueta del vértice v_i que mayor valor en la función Modularidad modificada presente.

Procedimentalmente, el algoritmo MOEA-SN comienza a partir de realizar la inicialización basada en la propagación de etiquetas. Seguidamente, a partir de la población inicial, se realiza un ordenamiento respecto a la dominancia, similar a NSGA-II, para luego aplicar operadores heurísticos de selección, cruzamiento y mutación, generándose una población descendiente. Por último, se efectúa un ordenamiento de las dos poblaciones, padres y descendientes, respecto a

la dominancia para seleccionar la próxima generación. En este paso se hace uso de la distancia apiñada. Para la obtención de la solución final, se escoge la solución del PS que mayor valor tiene en $(Q^+ + Q^-)$. Es utilizada NMI como métrica de calidad para la evaluación en los experimentos realizados.

4.2.4. Limitaciones o deficiencias presentes en los métodos descritos

Los métodos descritos anteriormente no contemplan explícitamente los pesos de los enlaces. En última instancia, estos enlaces son transformados a enlaces positivos o negativos. Los métodos reportados, si bien miden alta densidad de enlaces positivos dentro de la comunidad así como alta densidad negativa entre las comunidades, no contemplan medir baja densidad negativa dentro del grupo y baja densidad positiva entre los grupos, lo cual no permite identificar correctamente en las comunidades todas las relaciones presentes tanto dentro como fuera de esta.

Por otra parte, aunque bien es cierto que el método MODPSO minimiza tanto los enlaces positivos externos como los negativos internos, esta medición la realiza a nivel de la comunidad no a nivel de los vértices, representando un concepto de comunidad débil. Esto provoca que dos comunidades débiles, donde una tenga, a nivel de sus vértices, mayor enlaces negativos internos o positivos externos que la otra comunidad, serán vistas como comunidades similares. Tanto MEASN como MOEA-SN no incluyen minimizar los enlaces internos negativos como minimizar los enlaces externos positivos.

Por otra parte, todos los métodos son desarrollados en el enfoque estático, por lo cual serían muy ineficientes para procesar redes que sufren cambios en el tiempo. Adicionalmente, solo uno de los métodos es capaz de detectar comunidades con solapamiento, lo cual, como se dijo en la sección 4.1.13, refleja mejor lo que sucede en problemas reales. Por último, solo el método MODPSO implementa un mecanismo específico para identificar soluciones convergentes y diversas en el PF.

4.3. Algoritmos de detección de comunidades de redes dinámicas

Existen dos tipos de enfoques para el análisis estructural de las redes complejas. El enfoque estático se encarga de analizar los vértices y enlaces de la red en un momento dado con el objetivo de descubrir patrones estructurales en una configuración específica de los componentes de la red. Estas configuraciones se conocen como una *snapshot* o *instantánea* de la red. Por otro lado, el enfoque dinámico se centra en el estudio de redes que evolucionan en el tiempo; a este tipo de redes se les llama usualmente redes dinámicas. En este tipo de enfoque se analiza la red observada en múltiples instantes de tiempo y se centra en revelar los mecanismos evolutivos y los fenómenos dinámicos de la red que inciden en su estructura [45].

La mayoría de los algoritmos de detección de comunidades analizados en la sección 4.1 siguen un enfoque estático. Esto se debe principalmente, al hecho de que la detección de comunidades en redes estáticas sigue siendo un problema sumamente complejo, al cual se le dedica todavía esfuerzo. Con el perfeccionamiento de las tecnologías de la información, se ha incrementado la capacidad de generación y almacenamiento de redes etiquetadas temporalmente [45], lo cual ayuda a un mejor estudio de las redes dinámicas. Debido a que la mayoría de las redes reales están constantemente sometidas a cambios en su estructura, se ha evidenciado un creciente interés en el análisis estructural de las redes dinámicas.

En el contexto de la detección de comunidades en redes dinámicas desde un enfoque multi-objetivo, los métodos desarrollados y descritos en la presente sección, utilizan el enfoque de

agrupamiento evolutivo descrito en [91]. Además de desear ser capaz de descubrir eficazmente el conjunto de comunidades existente en un instante dado, este enfoque asume que las comunidades en dos instantes de tiempo t y $t + 1$ no cambian mucho; es decir, este enfoque asume que existe una estructura de comunidad la cual se debe mantener lo mas consistente posible a lo largo del tiempo. Con base a lo anterior, en este enfoque se define una función de costo que involucra: (1) la eficacia del agrupamiento en el instante t , denominado este término *snapshot cost* (SC, por sus siglas en inglés) y (2) la consistencia entre los agrupamientos en instantes de tiempo secuenciales, denominado *temporal cost* (TC, por sus siglas en inglés). Esta función de costo, denotada como $Cost$, queda definida de la siguiente forma [31]:

$$Cost = \alpha \times SC + (1 - \alpha) \times TC, \quad (31)$$

donde SC mide la eficacia de la estructura de comunidad S_t , identificada en el instante t y TC, calcula la similitud entre la estructura de comunidades S_t y la estructura S_{t-1} , identificada en el instante de tiempo anterior. El parámetro α , fijado por el usuario o calibrado automáticamente, controla el énfasis a realizar sobre la eficacia en el instante t o la consistencia de las estructuras a lo largo de los instantes de tiempo tomados.

El enfoque multi-objetivo en el contexto dinámico mediante el uso del concepto de [91], busca identificar un conjunto de soluciones que establezcan un compromiso entre las funciones SC y TC. En la presente sección, se describen varios algoritmos de detección de comunidades en un contexto dinámico siguiendo un enfoque multi-objetivo.

4.3.1. DYN-LSNNIA

El algoritmo DYN-LSNNIA (Dynamic multiobjective community detection algorithm based on NNIA with local search) [31], utiliza como metaheurísticas Nondominated Neighbor Immune Algorithm (NNIA, por sus siglas en inglés) definida en [92] y búsqueda local. A partir del uso del concepto de [91], DYN-LSNNIA propone utilizar la medida de Modularidad para medir la calidad de la comunidad, función SC, y la medida NMI para medir la similitud entre las comunidades en dos instantes de tiempo, función TC. Las soluciones son representadas mediante localización adyacente, lo cual permite la identificación automática del número de grupos. DYN-LSNNIA construye la población inicial a partir del uso de los nodos adyacentes, lo cual permite crear estructuras consistentes con la red.

El procedimiento toma como entrada una secuencia de T grafos $\{G_1, G_2, \dots, G_T\}$, siendo T el número de instantáneas o *snapshot* a contemplar. La salida constituye una secuencia de estructuras de comunidades $\{S_1, S_2, \dots, S_T\}$ identificadas en la red dinámica. DYN-LSNNIA construye la estructura de comunidades de G_1 , a partir del algoritmo propuesto por [14]. Luego, para cada instante t , DYN-LSNNIA construye el conjunto de soluciones no dominadas, que representan las posibles estructuraciones en comunidades que existen en la red en dicho instante de tiempo. Estas soluciones son mejoradas mediante una búsqueda local utilizando un enfoque de agregación; las soluciones mejoradas constituyen la generación inicial que utilizará DYN-LSNNIA cuando procese el grafo del próximo instante de tiempo. Para el proceso de búsqueda, DYN-LSNNIA reemplaza los operadores originales del método NNIA por el operador de cruzamiento uniforme y el operador de mutación en un punto aleatorio, utilizando los vecinos adyacentes. Ambos operadores se encuentran descritos en la sección 3.2.3.

La secuencia de estructuras de comunidades que devuelve DYN-LSNNIA se forma seleccionando, para cada instante t , a aquella solución del conjunto de soluciones no dominadas que alcance el mayor valor de la función CS. Es válido resaltar, que en el proceso de identificar la estructura

eficaz en el instante t solo es usado una función objetivo. De esta forma, el proceso de optimización multi-objetivo queda relegado solamente en la correlación existente entre eficacia obtenida en el instante t y poca variación entre las estructuras identificadas en instantes secuenciales. Además la propuesta no identifica estructuras de comunidades con solapamiento.

4.3.2. DYN-MOGA

El algoritmo DYN-MOGA (DYNamic MultiObjective Genetic Algorithms) [32], utiliza el mismo concepto dado por [91] y formula el problema de detección de comunidades dinámicas como un MOPs. De manera general, se busca que un objetivo maximice la calidad o eficacia de la estructura de comunidades en un instante t , mientras otro objetivo minimiza la diferencia que existe entre dos estructuras determinadas consecutivamente. En este sentido, DYN-MOGA utiliza varias funciones descritas en la sección 2.3 para medir el argumento definido como SC y NMI para medir el argumento TC.

DYN-MOGA utiliza una representación basada en localización adyacente y NSGA-II como método de solución. Para construir la población inicial, DYN-MOGA utiliza los vértices adyacentes de un nodo, de forma que se se tengan en cuenta los enlaces presentes en la red a la hora de formar las soluciones iniciales. La estructura del grafo correspondiente al instante $t = 1$ se determina mediante un GAs que utiliza solo la función establecida para medir SC. Luego, secuencialmente, se busca la estructura del grafo en el instante $t > 1$ mediante una búsqueda iterativa que utiliza el operador de cruzamiento uniforme y el operador de mutación aleatorio, acotado por los vértices adyacentes. Estos operadores han sido utilizados ampliamente por los métodos revisados en el presente trabajo.

La inicialización utiliza también la estructura topológica de la red para generar soluciones en correspondencia a los enlaces presentes en la red. Durante esta búsqueda se hace uso del ordenamiento de dominancia definido por el algoritmo NSGA-II. La próxima generación se construye tomando, de la jerarquía de fronteras de Pareto, aquellas soluciones de menor jerarquía. La solución final será aquella que alcance el mayor valor de Modularidad dentro del conjunto de soluciones no dominadas. DYN-MOGA es comparado con los algoritmos mono-objetivos propuestos en [93] y [94] en términos de la calidad de la estructuración, calculada con la medida NMI. Esta propuesta, al igual que la anterior, utiliza solamente una función objetivo en la identificación de la estructura en el instante t y construye comunidades disjuntas.

4.3.3. DYN-DMLS

El algoritmo DYN-DMLS (Decomposition-based MOEA for community detection in dynamic social networks) [33], al igual que los trabajos previos, formula el problema de detección de comunidades dinámicas como un MOPs y utiliza el concepto dado por [91]. A partir de una representación basada en localización adyacente, la optimización multi-objetivo se aborda con el método de solución MOEA/D descrito en la sección 3.2.2. La descomposición del problema multi-objetivo en varios problemas de optimización escalar se realiza mediante el método Tchebycheff, descrito en la ecuación (21). Las funciones utilizadas para medir el SC y el TC son la Modularidad y el NMI, respectivamente.

El proceso de búsqueda comienza identificando la estructura de comunidad del grafo perteneciente al instante t . DYN-DMLS construye las soluciones iniciales de manera similar a los métodos DYN-LSNNIA y DYN-MOGA. Así mismo, utiliza el operador de cruzamiento y operador de mutación que emplean los métodos antes mencionados. DYN-DMLS utiliza el método de propagación

de etiquetas descrito en [81] para lograr una optimización más eficaz y eficiente. Finalmente, para formar la secuencia de estructuras de comunidades, DYN-DMLS selecciona, para cada instante t , aquella solución dentro del conjunto de soluciones no dominadas, que alcance el mayor valor de Modularidad.

4.3.4. Limitaciones o deficiencias presentes en los métodos descritos

Los métodos analizados en las secciones anteriores solo contemplan adición y eliminación de aristas no así de vértices, lo cual en el contexto de las redes sociales puede ocurrir con mucha frecuencia. Además, estos métodos solo descubren comunidades disjuntas, lo cual, como se ha explicado, no permite reflejar con naturalidad lo que ocurre en los problemas reales. Por otra parte, los métodos descritos anteriormente solo utilizan en el proceso de optimización una función objetivo para medir la eficacia del agrupamiento en el instante t , por lo que queda sesgada la estructura de comunidades identificada, al tipo de propiedad que mide dicha función objetivo.

El concepto utilizado [91] por los métodos reportados está orientado a medir estabilidad o consistencia a los largo de la evolución de la red. Esto es útil en contextos donde se tiene definida la estructura a identificar, la cual aunque puede cambiar, los cambios son leves. Este concepto también es útil en dominios de aplicación donde el usuario consulta frecuentemente la estructura identificada. Por otra parte, en contextos donde lo que importa es cómo ha cambiado a lo largo del tiempo las comunidades, así como identificar qué transformaciones han sucedido en las estructuras, este concepto no es adecuado. Por último, es válido resaltar que los métodos descritos en la sección anterior no desarrollan un mecanismo específico para identificar soluciones convergentes y diversas en el PF, lo cual, como se ha mencionado, puede dificultar la convergencia del método al PF, así como la diversidad de las soluciones encontradas.

4.4. Algoritmos de detección de comunidades de redes multicaracterísticas

Muchos de los métodos desarrollados para el problema de detección de comunidades trabajan sobre redes que presentan un solo tipo de relaciones entre los vértices. Sin embargo, en la práctica, una red social puede estar caracterizada por diferentes tipos de relaciones entre sus miembros, como por ejemplo, relaciones laborales, relaciones familiares o relaciones de amistad, entre otras. Las redes que describen diferentes tipos de enlaces son conocidas como *redes multi-características* o *redes multi-dimensionales*. En la presente sección, se describe un método en el contexto multi-objetivo que aborda este tipo de red.

4.4.1. MultiMOGA

El algoritmo MultiMOGA (Multidimensional MultiObjective Genetic Algorithm) [29], persigue el objetivo de identificar, a partir de una red descrita por diferentes tipos de relaciones o dimensiones, una estructura de comunidad eficaz que tenga en cuenta todas las dimensiones de la red. Para ello, MultiMOGA redefine el concepto dado por [91] y asume un orden entre las dimensiones o tipos de relación. En este sentido, el concepto *temporal smoothness* se define como *dimensional sharing*, el concepto *snapshot quality* se define como *facet quality* (FQ, por sus siglas en inglés) y el concepto *temporal cost* como *sharing cost* (SC, por sus siglas en inglés). FQ mide la eficacia de la estructura de comunidades identificada a partir de la optimización de una función objetivo, teniendo en cuenta una de las relaciones presentes en la red, o sea, una dimensión. En el caso

de SC, mide cuán similar es la estructura determinada en la dimensión actual, respecto a la estructura identificada en la dimensión anterior.

El método utiliza una secuencia A_1, A_2, \dots, A_d de matrices adyacentes asociadas a los grafos $\{G_1, G_2, \dots, G_d\}$, donde d representa el número de características presentes en la red y cada grafo G_i , modela la red teniendo en cuenta a la característica i -ésima. A partir de lo anterior, el objetivo del método es identificar la estructura de comunidad que sea lo más consecuente posible con las redes modeladas para cada característica o dimensión, optimizando las funciones FQ y SC. De esta forma, la estructura de comunidad dada como solución en el procesamiento del Grafo G_d se considera como la estructuración más consistente a lo largo de todas las dimensiones o características. La función objetivo utilizada para medir FQ es la Modularidad. De forma similar, el índice o medida NMI se utiliza para medir SC; es decir, para saber cuánta similitud existe entre las estructuras identificadas en la dimensión actual y la identificada en la dimensión anterior.

MultiMOGA utiliza una representación basada en localización adyacente y conforma la estructura de comunidad en la primera dimensión mediante el uso de un GAs que utiliza la Modularidad como función objetivo. Luego, por cada dimensión $i > 1$, se construye el conjunto de Pareto optimizando FQ y SC, buscando así eficacia en las estructuras identificadas en el grafo G_i y consistencia entre las estructuras obtenidas en el grafo G_i y G_{i-1} . La construcción de las soluciones iniciales se realiza de forma similar a los métodos descritos en la sección previa, así como se utilizan los mismos operadores de cruzamiento y mutación. MultiMOGA selecciona como solución final aquella con mayor valor de Modularidad.

Como dos vértices pueden estar relacionados en una dimensión, no siendo así en el resto, existen vértices que no tienen enlaces en la dimensión actual (vértices aislados) y por tanto no son contemplados en la optimización del grafo G_i . Antes de iterar a la próxima dimensión, MultiMOGA realiza una asignación de etiqueta de grupo al nodo aislado mediante un procedimiento de propagación de etiquetas local. Este procedimiento, utiliza los vértices adyacentes del nodo aislado en todas las dimensiones y asigna la etiqueta que más se repite [29]. Al igual que los métodos anteriores, MultiMOGA no brinda un mecanismo específico para identificar soluciones convergentes y diversas en el PF. MultiMOGA construye estructuraciones que representan particiones duras.

4.4.2. Limitaciones o deficiencias presentes en los métodos descritos

MultiMOGA no contempla información descriptiva de los vértices que componen la red, solo se contemplan los diferentes tipos de enlaces, lo cual no permite identificar las comunidades utilizando toda la información que describe sus elementos. Además, el método no mide la importancia o significación de los diferentes tipos de relaciones en el proceso de búsqueda, sino que les asigna el mismo peso en la identificación de las estructuras las diferentes relaciones presentes en la red. Lo anterior no ilustra con naturalidad las relaciones presentes en una red social, en las cuales si pueden existir relaciones más importantes o definitorias a la hora de establecer vínculos entre individuos.

Por otro lado, MultiMOGA identifica estructuras de comunidades que representan particiones duras, lo cual tiene las limitantes comentadas en las secciones previas. Por último, al igual que los métodos anteriores, no se brinda un mecanismo específico para identificar soluciones convergentes y diversas en el PF.

5. Conclusiones

Como se pudo apreciar en el análisis presentado hasta el momento, los métodos desarrollados en su mayoría están orientados al contexto de las redes estáticas, evidenciándose muy poco trabajo en los contextos de redes con signo, redes dinámicas y redes multi-características. Respecto al tipo de estructura identificada, se observa que casi todos los algoritmos solo permiten detectar comunidades disjuntas, existiendo solamente tres trabajos que permiten construir comunidades con solapamiento. Como se comentó en secciones anteriores, las comunidades con traslape permiten reflejar de forma más natural lo que sucede en un problema real, en el cual, es común que los individuos pertenezcan a más de una comunidad, en dependencia de sus actividades o preferencias.

La mayoría de los métodos utilizan el concepto de comunidad descrito por [10], el cual solo describe información topológica de la red, además de presentar otras limitaciones enunciadas en la sección 4.1.13. Por otro lado, solo tres trabajos se han reportado que procesan cambios en la red, no obstante, estos no contemplan adiciones y/o eliminaciones de vértices, así como solo identifican comunidades disjuntas. Adicionalmente, el concepto de chakrabarti asume que existe una estructura de comunidades establecida que solo se puede modificar ligeramente, lo cual no es aplicable en dominios de aplicación donde sea de interés identificar y medir la evolución de la red.

Respecto a las redes con signo, solo tres trabajos se han reportado. Una de las limitantes de los métodos es no medir todas las relaciones positivas y/o negativas presentes en los vértices de la comunidad. En este sentido, la mayoría de los métodos solo tienen en cuenta para el análisis los enlaces positivos y negativos dentro de la comunidad y fuera de esta, respectivamente. De esta forma, no se contempla identificar poca densidad de enlaces negativos dentro de la comunidad, así como poca densidad de enlaces positivos fuera de esta. Por otro lado, estos trabajos no hacen un uso explícito de los pesos positivos o negativos presentes en la red, en última instancia, estos pesos son transformados a etiquetas que representan enlaces positivos o negativos. Por tanto, se excluye información útil que permita medir con mayor rigor las relaciones negativas y/o positivas entre los vértices de la red. Por último, estos trabajos solo contemplan un tipo de relación entre los vértices, ya sea negativa o positiva; de tal forma que los vértices solo pueden ser relacionados de manera positiva o negativa y no de ambas.

En el caso de las redes con diferentes tipos de relaciones entre sus vértices, solo un trabajo se ha reportado y en este solo se considera características de las aristas y no de los vértices, lo cual puede ocurrir en problemas reales. Tener en cuenta la información descriptiva tanto de los vértices como de sus relaciones, permitiría identificar estructuras de comunidades más acordes al problema a resolver.

La mayoría de los trabajos reportados utilizan algoritmos metaheurísticos como método de solución de problemas, sin embargo no se realiza un análisis riguroso en todos los casos respecto a la convergencia y/o diversidad de las soluciones pertenecientes a la frontera de Pareto. La falta de este análisis puede dificultar orientar la búsqueda hacia soluciones que converjan adecuadamente al frente de Pareto y a la vez tengan diversidad. Adicionalmente, en la mayoría de los trabajos no se tiene en cuenta las características de red para construir un conjunto de soluciones iniciales que sea prometedora.

Por otro lado, salvo MOCD, el resto de los métodos utilizan un índice de validación interno para la selección de la solución final. Este método de selección tiene el inconveniente descrito en la sección 4.1.13, donde el índice utilizado en la evaluación puede medir propiedades diferentes

a las identificadas por el algoritmo, lo cual provoca que las estructuras identificadas, podrían no dar valores correctos. Se evidencia el no uso de los métodos propuestos por [5] y descritos en las secciones 3.3.2 y 3.3.3, a pesar de ser métodos que contemplan todas las soluciones pertenecientes al conjunto de Pareto para la selección de la solución final. Por último, salvo el método MEA-CDPs, el resto solo optimiza dos funciones objetivos. Por tanto, no se logra medir al unísono otras propiedades útiles en las comunidades.

Con base en todo lo dicho hasta el momento, se puede concluir que la detección de comunidades siguiendo un enfoque de optimización multi-objetivo, es todavía un tema de interés de la comunidad científica.

Referencias bibliográficas

1. Zhou, Y., Wang, J., Luo, N., Zhang, Z.: Multiobjective local search for community detection in networks. *Soft Computing* (2015) 1–10
2. Pizzuti, C.: A multiobjective genetic algorithm to find communities in complex networks. *Evolutionary Computation, IEEE Transactions on* **16**(3) (2012) 418–430
3. Deb, K.: *Multi-objective optimization using evolutionary algorithms*. Volume 16. John Wiley & Sons (2001)
4. Shi, C., Yan, Z., Cai, Y., Wu, B.: Multi-objective community detection in complex networks. *Applied Soft Computing* **12**(2) (2012) 850–859
5. Zhang, Y., Zhang, X., Tang, J., Luo, B.: Decision-making strategies for multi-objective community detection in complex networks. In: *Bio-Inspired Computing-Theories and Applications*. Springer (2014) 621–628
6. Wasserman, S., Faust, K.: *Social network analysis: Methods and applications*. Volume 8. Cambridge university press (1994)
7. Lancichinetti, A., Fortunato, S.: Consensus clustering in complex networks. *Scientific reports* **2** (2012)
8. Pons, S.V., Ruiz-Shulcloper, J.: *Combinación de agrupamientos: un estado del arte*. Technical report, Dpto. Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba (2010)
9. Gandón, A.G., Pons, S.V., Ruiz-Shulcloper, J.: *Algoritmos de agrupamiento conceptuales: un estado del arte*. Technical report, Dpto. Reconocimiento de Patrones, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba (2012)
10. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America* **101**(9) (2004) 2658–2663
11. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Physical review E* **69**(2) (2004) 026113
12. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. *Physical review E* **72**(2) (2005) 027104
13. Tasgin, M., Herdagdelen, A., Bingol, H.: Community detection in complex networks using genetic algorithms. *arXiv preprint arXiv:0711.0491* (2007)
14. Pizzuti, C.: Ga-net: A genetic algorithm for community detection in social networks. In: *Parallel Problem Solving from Nature-PPSN X*. Springer (2008) 1081–1090
15. Shi, C., Yan, Z., Wang, Y., Cai, Y., Wu, B.: A genetic algorithm for detecting communities in large-scale complex networks. *Advances in Complex Systems* **13**(01) (2010) 3–17
16. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proceedings of the national academy of sciences* **99**(12) (2002) 7821–7826
17. Shi, C., Yu, P.S., Yan, Z., Huang, Y., Wang, B.: Comparison and selection of objective functions in multiobjective community detection. *Computational Intelligence* **30**(3) (2014) 562–582
18. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: *Proceedings of the 19th international conference on World wide web, ACM* (2010) 631–640
19. Kannan, R., Vempala, S., Vetta, A.: On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)* **51**(3) (2004) 497–515
20. Rosvall, M., Bergstrom, C.T.: An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences* **104**(18) (2007) 7327–7331
21. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. *Proceedings of the National Academy of Sciences* **104**(1) (2007) 36–41

22. Kumpula, J.M., Saramäki, J., Kaski, K., Kertész, J.: Limited resolution and multiresolution methods in complex network community detection. In: SPIE Fourth International Symposium on Fluctuations and Noise, International Society for Optics and Photonics (2007) 660116–660116
23. Shi, C., Cai, Y., Yu, P.S., Yan, Z., Wu, B.: A comparison of objective functions in network community detection. In: Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, IEEE (2010) 1234–1241
24. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* **1**(1) (2011) 32–49
25. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hofer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on* **20**(2) (2008) 172–188
26. Gong, M., Ma, L., Zhang, Q., Jiao, L.: Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Physica A: Statistical Mechanics and its Applications* **391**(15) (2012) 4050–4060
27. Gong, M., Cai, Q., Chen, X., Ma, L.: Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *Evolutionary Computation, IEEE Transactions on* **18**(1) (2014) 82–97
28. Zeng, Y., Liu, J.: Community detection from signed social networks using a multi-objective evolutionary algorithm. In: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1, Springer (2015) 259–270
29. Amelio, A., Pizzuti, C.: Community detection in multidimensional networks. In: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, IEEE (2014) 352–359
30. Bothorel, C., Cruz, J.D., Magnani, M., Micenkova, B.: Clustering attributed graphs: models, measures and methods. *Network Science* (2015) 1–37
31. Gong, M.G., Zhang, L.J., Ma, J.J., Jiao, L.C.: Community detection in dynamic social networks based on multiobjective immune algorithm. *Journal of Computer Science and Technology* **27**(3) (2012) 455–467
32. Folino, F., Pizzuti, C.: An evolutionary multiobjective approach for community discovery in dynamic networks. *Knowledge and Data Engineering, IEEE Transactions on* **26**(8) (2014) 1838–1852
33. Ma, J., Liu, J., Ma, W., Gong, M., Jiao, L.: Decomposition-based multiobjective evolutionary algorithm for community detection in dynamic social networks. *The Scientific World Journal* **2014** (2014)
34. Talbi, E.G.: *Metaheuristics from desing to implementation*. John Wiley & Sons (2009)
35. Agrawal, R.: Bi-objective community detection (bocd) in networks using genetic algorithm. In: *Contemporary Computing*. Springer (2011) 5–15
36. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. *Physical Review E* **74**(1) (2006) 016110
37. Li, Z., Zhang, S., Wang, R.S., Zhang, X.S., Chen, L.: Quantitative function for community detection. *Physical review E* **77**(3) (2008) 036109
38. Arenas, A., Fernández, A., Gómez, S.: Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics* **10**(5) (2008) 053039
39. Pons, P., Latapy, M.: Post-processing hierarchical community structures: Quality improvements and multi-scale view. *Theoretical Computer Science* **412**(8) (2011) 892–900
40. Angelini, L., Boccaletti, S., Marinazzo, D., Pellicoro, M., Stramaglia, S.: Identification of network modules by optimization of ratio association. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **17**(2) (2007) 023114
41. Gong, M., Chen, X., Ma, L., Zhang, Q., Jiao, L.: Identification of multi-resolution network structures with multi-objective immune algorithm. *Applied Soft Computing* **13** (2013) 1705–1717
42. Wei, Y.C., Cheng, C.K.: Ratio cut partitioning for hierarchical designs. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* **10**(7) (1991) 911–921
43. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* **11**(3) (2009) 033015
44. Demir, G.N., Uyar, A.S., Ögüdücü, S.G.: Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM (2007) 1943–1950
45. Alvarez, G.L., Pérez, A.S.: Detección de comunidades con traslape en redes complejas: Un estado del arte. Technical report, Dpto. Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba (2015)
46. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. *Evolutionary Computation, IEEE Transactions on* **11**(1) (2007) 56–76
47. Shi, C., Yu, P.S., Cai, Y., Yan, Z., Wu, B.: On selection of objective functions in multi-objective community detection. In: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM (2011) 2301–2304
48. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on* **11**(6) (2007) 712–731

49. Jaszkiwicz, A., Branke, J.: Interactive multiobjective evolutionary algorithms. In: *Multiobjective Optimization*. Springer (2008) 179–193
50. Talbi, E.G., Basseur, M., Nebro, A.J., Alba, E.: Multi-objective optimization using metaheuristics: non-standard algorithms. *International Transactions in Operational Research* **19**(1-2) (2012) 283–305
51. Kim, K., McKay, R.L., Moon, B.R.: Multiobjective evolutionary algorithms for dynamic social network clustering. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, ACM (2010) 1179–1186
52. Amiri, B., Hossain, L., Crawford, J.: A multiobjective hybrid evolutionary algorithm for clustering in social networks. In: *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, ACM (2012) 1445–1446
53. Li, Y., Wang, Y., Chen, J., Jiao, L., Shang, R.: Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization. *Journal of Heuristics* (2015) 1–27
54. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: A survey of multiobjective evolutionary clustering. *ACM Computing Surveys* **47**(4) (2015) 61:1–61:46
55. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., De Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. *Systems, Man, and Cybernetics - Part C: Applications and Reviews*, *IEEE Transactions on* **39**(2) (2009) 133–155
56. Park, Y.J., Song, M.S.: A genetic algorithm for clustering problems. In: *Proc. 3rd Annu. Conf. Genet. Algorithms*, IEEE (1998) 2–9
57. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation*, *IEEE Transactions on* **6**(2) (2002) 182–197
58. Weise, T.: *Global optimization algorithms-theory and application*. Self-Published, (2009)
59. Amiri, B., Hossain, L., Crawford, J.W., Wigand, R.T.: Community detection in complex networks: Multi-objective enhanced firefly algorithm. *Knowledge-Based Systems* **46** (2013) 1–11
60. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* **2005**(09) (2005) P09008
61. Liu, J., Zhong, W., Abbass, H., Green, D.G., et al.: Separated and overlapping community detection in complex networks using multiobjective evolutionary algorithms. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*, IEEE (2010) 1–7
62. Liu, B., Wang, C., Wang, C.: Multiobjective genetic method for community discovery in complex networks. In: *Advances in Swarm Intelligence*. Springer (2014) 404–413
63. Newman, M.E.: Fast algorithm for detecting community structure in networks. *Physical review E* **69**(6) (2004) 066133
64. Pizzuti, C.: A multi-objective genetic algorithm for community detection in networks. In: *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, IEEE (2009) 379–386
65. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. *Physical review E* **70**(6) (2004) 066111
66. Pons, P., Latapy, M.: Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **10**(2) (2006) 191–218
67. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10) (2008) P10008
68. Demir, G.N., Uyar, A.Ş., Gündüz-Öğüdücü, Ş.: Multiobjective evolutionary clustering of web user sessions: a case study in web page recommendation. *Soft Computing* **14**(6) (2010) 579–597
69. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation* **8**(2) (2000) 173–195
70. Bandyopadhyay, S., Pal, S.K., Aruna, B.: Multiobjective gas, quantitative indices, and pattern classification. *Systems, Man, and Cybernetics, Part B: Cybernetics*, *IEEE Transactions on* **34**(5) (2004) 2088–2099
71. Pujol, J.M., Béjar, J., Delgado, J.: Clustering algorithm for determining community structure in large networks. *Physical Review E* **74**(1) (2006) 016107
72. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J., et al.: Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, Citeseer (2001)
73. Gong, M., Fu, B., Jiao, L., Du, H.: Memetic algorithm for community detection in networks. *Physical Review E* **84**(5) (2011) 056101
74. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* **105**(4) (2008) 1118–1123
75. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**(2) (2001) 60–68

76. Choi, C., Lee, J.J.: Chaotic local search algorithm. *Artificial Life and Robotics* **2**(1) (1998) 41–47
77. Yang, X.S.: Firefly algorithms for multimodal optimization. In: *Stochastic algorithms: foundations and applications*. Springer (2009) 169–178
78. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on* **7**(3) (2003) 289–304
79. Golberg, D.E.: *Genetic algorithms in search, optimization, and machine learning*. Addison wesley **1989** (1989)
80. Hafez, A.I., Al-Shammari, E.T., ella Hassanien, A., Fahmy, A.A.: Genetic algorithms for multi-objective community detection in complex networks. In: *Social Networks: A Framework of Computational Intelligence*. Springer (2014) 145–171
81. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* **76**(3) (2007) 036106
82. Sun, J., Feng, B., Xu, W.: Particle swarm optimization with particles having quantum behavior. In: *Congress on Evolutionary Computation*. (2004)
83. Xi, M., Sun, J., Xu, W.: An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. *Applied Mathematics and Computation* **205**(2) (2008) 751–759
84. Coello, C.A.C., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. *Evolutionary Computation, IEEE Transactions on* **8**(3) (2004) 256–279
85. Pizzuti, C.: Overlapped community detection in complex networks. In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM* (2009) 859–866
86. Liu, C., Liu, J., Jiang, Z.: A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *Cybernetics, IEEE Transactions on* **44**(12) (2014) 2274–2287
87. Huang, J., Sun, H., Liu, Y., Song, Q., Weninger, T.: Towards online multiresolution community detection in large-scale networks. *PloS one* **6**(8) (2011) e23829
88. Tang, J., Lou, T., Kleinberg, J.: Inferring social ties across heterogenous networks. In: *Proceedings of the fifth ACM international conference on Web search and data mining, ACM* (2012) 743–752
89. Goldberg, D.E., Lingle, R.: Alleles, loci, and the traveling salesman problem. In: *Proceedings of the first international conference on genetic algorithms and their applications, Lawrence Erlbaum Associates, Publishers* (1985) 154–159
90. Shi, C., Yan, Z., Wang, Y., Cai, Y., Wu, B.: Genetic algorithm with local search for community detection in large-scale complex networks. *Advances in Complex Systems* **13**(1) (2010) 873–882
91. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM* (2006) 554–560
92. Gong, M., Jiao, L., Du, H., Bo, L.: Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation* **16**(2) (2008) 225–255
93. Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **3**(2) (2009) 8
94. Kim, M.S., Han, J.: A particle-and-density based evolutionary clustering method for dynamic networks. *Proceedings of the VLDB Endowment* **2**(1) (2009) 622–633

RT_034, febrero 2016

Aprobado por el Consejo Científico CENATAV

Derechos Reservados © CENATAV 2016

Editor: Lic. Lucía González Bayona

Diseño de Portada: Di. Alejandro Pérez Abraham

RNPS No. 2143

ISSN 2072-6260

Indicaciones para los Autores:

Seguir la plantilla que aparece en www.cenatav.co.cu

C E N A T A V

7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;

La Habana. Cuba. C.P. 12200

Impreso en Cuba

