PLAN DE INVESTIGACIÓN DE BECA UBACYT 2023
Categorías Maestría/Doctorado/Culminación

Apellido y Nombre	Brandwein Eric
DNI	40392482
Unidad Académica	Ciencias Exactas y Naturales

1. Título del Plan de Investigación:

Parámetros de ancho y algoritmia en grafos

CTA	**Ingenierías, Ciencias del Ambier**
Disciplina	**Cs. de la Computación**
Rama	**Teoría de Grafos**

2. Resumen en español (hasta 200 palabras)

Este plan se enfoca en complejidad parametrizada, en relación a los parámetros de ancho conocidos como *thinness* y *treewidth*. En pocas palabras, complejidad parametrizada es un marco para analizar cómo una medida secundaria de una instancia, llamada el *parámetro*, afecta el tiempo de ejecución de un algoritmo, por encima de la medida convencional definida por el tamaño de una instancia.

Los parámetros de ancho desempeñan un rol fundamental en la teoría estructural y algorítmica de grafos. La *thinness* es uno de los pocos parámetros que resulta acotado para grafos de intervalos, y por lo tanto permite generalizar sus propiedades algorítmicas.

Se conocen algoritmos XP parametrizados por thinness para distintos problemas de optimización combinatoria, incluyendo clique máxima, conjunto independiente máximo, conjunto dominante mínimo.

El problema de decidir si la *thinness* de un grafo es a lo sumo *k* es NP-completo, pero la complejidad no se conoce para *k* constante al menos 2. Como parte de este plan, se estudiará la existencia de algoritmos polinomiales restringidos a una cierta clase, o parametrizados ya sea por la misma *thinness* o por otros invariantes, como el *treewidth* o el *vertex cover*. También se busca obtener cotas inferiores de complejidad bajo supuestos como la ETH.

3. Título y resumen en inglés (hasta 200 palabras)

Title: Width parameters and algorithms on graphs.

This plan focuses on parameterized complexity, in relation to the width parameters known as thinness and treewidth. In a nutshell, parameterized complexity is a framework for analyzing how a secondary measure of an instance, called the parameter, affects the running time of an algorithm, over and above the conventional measure defined by the size of an instance.

The width parameters play a fundamental role in the structural and algorithmic graph theory. The thinness is one of the few parameters that is bounded for interval graphs, and therefore allows generalizing their algorithmic properties.

Thinness-parameterized XP algorithms are known for different combinatorial optimization problems, including maximum clique, maximum independent set, and minimum dominant set.

The problem of deciding whether the thinness of a graph is at most *k* is NP-complete, but the complexity is not known for *k* constant at least 2. As part of this plan, we will study the existence of polynomial algorithms restricted to certain classes, or parameterized either by the thinness itself or by other invariants, such as treewidth or vertex cover. We also seek to obtain complexity lower bounds under assumptions such as ETH.

4. Estado actual del conocimiento sobre el tema (desarrollar en no más de 900 palabras)

Los algoritmos son procedimientos para resolver problemas computacionales. Desde el punto de vista teórico, el diseño de algoritmos eficientes debe ir de la mano con un análisis riguroso basado en un marco matemático.

Desafortunadamente, la mayoría de los problemas de optimización combinatoria que surgen tanto en la teoría como en la práctica resultan ser NP-completos. Al manejar tales problemas, a menos que P=NP, no podemos esperar un algoritmo de tiempo *polinomial* que *siempre* produce una solución *óptima* para *cualquier* instancia. En la actualidad existen varios paradigmas para eludir ese límite intrínseco de computabilidad, basados en relajar esos requisitos (eficiencia, determinismo, optimalidad, generalidad), tratando de optimizar el rendimiento en otros aspectos. Por ejemplo, el desarrollo de marcos para el diseño y análisis de algoritmos de aproximación, aleatorizados, moderadamente exponenciales, polinomiales para conjuntos de instancias particulares [6], o el enfoque de *complejidad parametrizada* [8,9,11], que es el enfoque principal de este plan de beca. En pocas palabras, complejidad parametrizada es un marco para analizar cómo una medida secundaria de una instancia, llamada el *parámetro*, afecta el tiempo de ejecución de un algoritmo, por encima de la medida convencional definida por el tamaño de una instancia. La idea central de la complejidad parametrizada es refinar la visión de complejidad agregando una o más dimensiones para medir la "dureza" de un problema.

Más precisamente, la complejidad parametrizada se ocupa de problemas y algoritmos cuya complejidad es medida no solamente basándose en el tamaño de la instancia, sino también en un valor entero k asociado a otra característica de la instancia, llamado el *parámetro*. Así, algoritmos que resuelven tal problema con complejidad temporal polinomial en el tamaño de la entrada al fijar como constante el parámetro, se dice que pertenecen a la clase de complejidad XP, o *slicewise polynomial* (complejidad $n^{f(k)}$, donde f es alguna función computable, n el tamaño de la instancia, y k el parámetro). Otra clase de complejidad más restrictiva, a la que apuntaremos con nuestros resultados algorítmicos, es la clase FPT, o *fixed parameter tractable*, en la que la complejidad requerida para los algoritmos es $f(k) \cdot poli(n)$, donde f es alguna función computable, *poli* un polinomio, n el tamaño de la instancia, y k el parámetro.

Los parámetros de ancho desempeñan un rol fundamental en la teoría estructural y algorítmica de grafos, y la jerarquía de parámetros definidos crece año a año. En términos generales, tales parámetros son invariantes que cuantifican qué tan "bien" se puede descomponer un grafo en una estructura de tipo árbol o camino. Entre ellos, el *treewidth* es probablemente el más relevante. Fue introducido por Robertson y Seymour en 1984 con fines estructurales en su proyecto seminal *Graph Minors* y desde entonces se ha convertido en una de las herramientas más útiles en el diseño de algoritmos parametrizados [15]. La *thinness* de un grafo es otro de los tantos parámetros de ancho de un grafo, definido por Mannino, Oriolo, Ricci y Chandran en 2007 [14], que está cobrando cada vez más relevancia. Es uno de los pocos parámetros que resulta acotado para grafos de intervalos, y por lo tanto permite generalizar las tantas propiedades algorítmicas de esa clase de grafos.

Formalmente, un grafo es ***k*-thin** si existe un orden de sus vértices y una partición de los mismos en *k* clases tal que para todo v < w < z vértices, si v y w pertenecen a la misma clase y vz es una arista del grafo, entonces también lo es wz. La ***thinness*** de un grafo es el menor *k* tal que es *k*-thin.

Tanto para el *treewidth* como para la *thinness* se conocen algoritmos parametrizados para problemas de optimización combinatoria, incluyendo algunos de los más famosos, como clique máxima, conjunto independiente máximo, conjunto dominante mínimo. En general, los algoritmos parametrizados por *treewidth* son FPT [15], mientras que en el caso de la *thinness*, los algoritmos que se conocen son XP [4]. Más aún, el problema de coloreo es NP-completo para grafos de *thinness* 2 [2].

El problema de decidir si la *thinness* de un grafo es a lo sumo *k* es NP-completo [16]. Es sabido que los grafos de intervalos (equivalentes a los grafos de *thinness* 1) se pueden reconocer en tiempo polinomial [12], pero la complejidad del problema de reconocimiento de grafos de *thinness* *k* permanece abierto para *k* constante mayor o igual a 2.

Resulta relevante estudiar si el problema admite algoritmos polinomiales cuando se restringe el grafo de entrada a una cierta clase, y si el problema admite algoritmos polinomiales parametrizados ya sea por la misma *thinness* o por otros invariantes, como ser el *treewidth* o el *vertex cover*, que también es frecuentemente utilizado como parámetro para el cálculo de parámetros de ancho, entre otros problemas [7,10]. Como parte de la tesis de licenciatura del postulante a esta beca (publicada en [1]), se presentó un algoritmo polinomial para calcular la *thinness* de un bosque, y en [4] se computa la *thinness* de los cografos en tiempo polinomial. Intuimos que el problema parametrizado por *treewidth* resultará más desafiante que por *vertex cover*, ya que los árboles (*treewidth* 1) tienen *thinness* no acotada y su cómputo en árboles es no trivial, mientras que el *vertex cover* es una cota superior para la *thinness* de un grafo [5].

También se utilizará el concepto de *fine grained complexity* para obtener cotas inferiores de complejidad bajo supuestos razonables y ampliamente aceptados, como la ETH [13].

5. Objetivos e hipótesis del Plan de Trabajo a realizar (desarrollar en no más de 900 palabras)

Yaroslav Shitov demostró en 2021 que decidir si un grafo tiene *thinness* a lo sumo *k* es NP-completo. Sin embargo, la complejidad está abierta cuando *k* es constante (salvo en el caso *k*=1 [12]), lo cual lleva a los primeros objetivos:

1. Encontrar un algoritmo eficiente para decidir si un grafo tiene *thinness* a lo sumo *k*, donde *k* es constante. Alternativamente, demostrar que dicho problema es NP-completo.

2. De no ser posible encontrar el algoritmo del objetivo 1, estudiar la existencia de un algoritmo eficiente parametrizado por *k* tal que, dado un grafo, si el grafo tiene *thinness* *k*, obtenga una representación de ancho a lo sumo *f(k)* (*f* función fija).

3. De no ser posible encontrar el algoritmo del objetivo 1, estudiar la existencia de un algoritmo eficiente parametrizado por *k* tal que, dada una representación de ancho a lo sumo *k* de un grafo, calcule la *thinness* exacta del grafo.

Un bosque es un grafo sin ciclos, o en otras palabras, un grafo tal que cada una de sus componentes conexas es un árbol. Como parte de la tesis de licenciatura del postulante a esta beca (publicada posteriormente en [1]), se presentó un algoritmo para encontrar la thinness de un bosque en tiempo polinomial. Los bosques no triviales son exactamente los grafos de *treewidth* 1, y por lo tanto de aquí surge el siguiente objetivo del plan:

4. Encontrar un algoritmo eficiente para calcular la *thinness* de un grafo, parametrizado por su *treewidth*, o determinar que no existe tal algoritmo bajo suposiciones de complejidad razonables (ej. [13]). De no ser posible, encontrar algoritmos eficientes para valores fijos de *treewidth*, comenzando por los grafos de *treewidth* 2.

En esa tesis también, como corolario del algoritmo, se obtuvieron cotas para la *thinness* de un árbol en función de su cantidad de hojas. Un resultado similar vale para los grafos cordales, donde la *thinness* resulta estar acotada por la *leafage* menos 1 [3]. Los grafos de intervalo son los cordales de *leafage* 2. Eso lleva al siguiente objetivo:

5. Encontrar un algoritmo eficiente para calcular la *thinness* de un grafo cordal, parametrizado por su *leafage*, o determinar que no existe tal algoritmo bajo suposiciones de complejidad razonables. De no ser posible, encontrar algoritmos eficientes para valores fijos de *leafage*, comenzando por los grafos cordales de *leafage* 4 (para los de *leafage* 3, se sabe que tienen *thinness* a lo sumo 2, y se puede reconocer en tiempo polinomial si tienen *thinness* 1).

También se sabe que en grafos de co-comparabilidad, la *thinness* está acotada por el número cromático [4]. Planteamos dos objetivos relacionados con eso:

6. Encontrar un algoritmo eficiente para calcular la *thinness* de un grafo de co-comparabilidad, parametrizado por su número cromático, o determinar que no existe tal algoritmo bajo suposiciones de complejidad razonables.
7. Mejorar la cota conocida para grafos de co-comparabilidad, usando algún otro parámetro o combinación de parámetros, ya que esa cota vale por igualdad en algunos casos, pero puede ser arbitrariamente mala para, por ejemplo, grafos de intervalos.

El *vertex cover* es un parámetro que ha resultado útil a la hora de encontrar algoritmos parametrizados para el cálculo de otros parámetros de ancho [7], y es una cota superior para la *thinness* de un grafo (no vacío) [5]. Un objetivo del plan es el siguiente:

8. Encontrar un algoritmo eficiente para calcular la *thinness* de un grafo, parametrizado por su *vertex cover*, o su *modular width*. En este caso, tenemos indicios de que tal algoritmo existe, al menos para el *vertex cover*.

En la tesis de licenciatura de Diego de Estrada, publicada en [4], se presentaron algoritmos para resolver una gran cantidad de problemas conocidos en tiempo polinomial cuando la *thinness* del grafo de entrada es acotada. Los algoritmos tienen complejidad en XP parametrizados por la *thinness*. El siguiente objetivo se centra en esta familia de problemas:

9. Encontrar algoritmos de complejidad FPT para algunos problemas de esa familia, o determinar que no existen tales algoritmos bajo suposiciones de complejidad razonables. En ese caso, tratar de encontrar algoritmos XP más eficientes para resolver los mismos problemas en grafos de *thinness* acotada.

6. Metodología a aplicar y adecuación con el objeto de estudio, la temática y los objetivos (desarrollar en no más de 900 palabras)

Se comenzará por cursar materias relacionadas al tema, incluyendo materias de grafos y de algoritmia. Por ser el tema propuesto un tema que no es abarcado por las materias obligatorias de grado en la carrera, la primera parte del doctorado consistirá en ponerse al día con los últimos avances sobre la materia. Se leerá bibliografía adecuada (hay varios libros recientes sobre complejidad parametrizada, y un gran número de papers como para incorporar las técnicas típicamente utilizadas en esta clase de algoritmos) para poder avanzar en los objetivos propuestos, y se propondrán distintas posibles ramas de investigación. El punto de partida serán los objetivos propuestos en este plan y se irán adecuando a lo largo de los años las preguntas a los nuevos resultados obtenidos, ya sea por el postulante o por la comunidad internacional. Siendo un área de investigación muy activa, se intentará participar asiduamente de workshops y congresos en el tema.

Para algunos de estos primeros objetivos ya tenemos alguna idea clara de un camino a la solución (por ejemplo, para el algoritmo que calcule la *thinness* de un grafo parametrizado por su *vertex cover number*). Para otros, contamos con los últimos avances en problemas relacionados, pero requieren ser pensados y que surjan ideas originales para resolverlos, y en algunos podrían terminar siendo demasiado difíciles. Es por eso que el número de objetivos es grande, y de hecho se espera que la tesis termine teniendo resultados relacionados con algunos de ellos, no necesariamente todos.

Se investigará en las oficinas proporcionadas por la UBA, junto a la directora de doctorado Flavia Bonomo. Además, se realizarán reuniones periódicas con el co-director Ignasi Sau, HDR en el LIRMM, Université de Montpellier, Montpellier, Francia. Algunas se realizarán por video llamada, y otras en persona, a través de visitas mutuas financiadas por proyectos de cooperación o por el Laboratorio Asociado Franco-Argentino SINFIN.

7. Cronograma de actividades (consignar sucesivamente cada actividad unitaria, según corresponda)

Actividad	Año 1											
	1	2	3	4	5	6	7	8	9	10	11	12
Cursada de Materias			x	x	x	x	x	x	x	x	x	
Lectura de bibliografía	x	x	x	x	x	x	x	x	x	x	x	x
Investigación (Objetivos 8, 5, 9)	x	x	x	x	x	x	x	x	x	x	x	x
Participación en workshops				x	x				x	x	x	
Actividad	Año 2											
	1	2	3	4	5	6	7	8	9	10	11	12
Cursada de Materias			x	x	x	x	x	x	x	x	x	
Lectura de bibliografía	x	x	x	x	x	x	x	x	x	x	x	x
Investigación (Objetivos 4, 6, 7, 9)	x	x	x	x	x	x	x	x	x	x	x	x
Participación en congresos y workshops				x	x				x	x	x	
Actividad	Año 3 *											
	1	2	3	4	5	6	7	8	9	10	11	12
Investigación (Objetivos 1, 2, 3, 9)	x	x	x	x	x	x	x	x	x	x	x	x
Publicación de papers	x	x	x	x	x	x	x	x	x	x	x	x
Participación en congresos y workshops				x	x				x	x		
Escritura de la tesis	x	x	x	x	x	x	x	x	x	x	x	x

* Deberá completarse en caso de presentarse a una Beca Doctorado

8. Bibliografía (citada y consultada, desarrollar en no más de una carilla)

1. Bonomo F., Brandwein E., González C.L., and Sansone A., **On the Thinness of Trees**, ISCO 2022, LNCS 13526: 189-200 (2022).
2. Bonomo F., Brandwein E., Oliveira F.S., Sampaio Jr. M.S., Sansone A., and Szwarcfiter J.L.: **Thinness and its variations on some graph families and coloring graphs of bounded thinness**, enviado (2023). arXiv: 2303.06070
3. Bonomo F., Brettell N., Munaro A., and Paulusma D., **Classifying the Thinness of H-Graphs**, manuscrito (2021).
4. Bonomo F. and De Estrada D., **On the thinness and proper thinness of a graph**, Discrete Applied Mathematics 261: 78-92 (2019).
5. Bonomo F., Gonzalez C.L., Oliveira F.S., Sampaio Jr. M.S., and Szwarcfiter J.L.: **Thinness of product graphs**, Discrete Applied Mathematics 312: 52-71 (2022).
6. Brandstädt A., Le V.B., and Spinrad J.P., **Graph Classes: A Survey**, SIAM, 1999.
7. Chapelle M., Liedloff M., Todinca I., and Villanger Y., **Treewidth and Pathwidth parameterized by the vertex cover number**, Discrete Applied Mathematics 216(1): 114-129 (2017).
8. Cygan M., Fomin F.V., Kowalik L., Lokshtanov D., Marx D., Pilipczuk M., Pilipczuk M., and Saurabh S., **Parameterized Algorithms**, Springer, 2015.
9. Downey R.G. and Fellows M.R., **Fundamentals of parameterized complexity**, Springer, 2013.
10. Fomin F.V., Liedloff M., Montealegre P., and Todinca I., **Algorithms Parameterized by Vertex Cover and Modular Width, Through Potential Maximal Cliques**, Algorithmica 80: 1146–1169 (2018).
11. Garnero V., Paul C., Sau I., and Thilikos D.M., **Explicit Linear Kernels via Dynamic Programming**, SIAM Journal on Discrete Mathematics 29(4): 1864-1894 (2015).
12. Habib M., McConnell R., Paul C., and Viennot L., **Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing**, Theoretical Computer Science 234(1-2): 59-84 (2000).
13. Lokshtanov D., Marx D., and Saurabh S., **Lower bounds based on the exponential time hypothesis**, Bulletin of the EATCS, (105): 41–72 (2011).
14. Mannino C., Oriolo G., Ricci F., and ChandranS., **The stable set problem and the thinness of a graph**, Operations Research Letters 35: 1-9 (2007).
15. Robertson N. and Seymour P.D., **Graph Minors. II. Algorithmic Aspects of Tree-Width**, Journal of Algorithms 7(3): 309-322 (1986).
16. Shitov Y.: **Graph thinness is NP-complete**, manuscrito (2021).