



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS Y
DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

TRANSICIÓN DE ENCIERRO EN CADENAS POLIGONALES
TRIDIMENSIONALES

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO (A) EN CIENCIAS

PRESENTA:
Aldo Sayeg Pasos Trejo

DIRECTOR
Dr. Ricardo Atahualpa Solórzano Kraemer

CIUDAD DE MÉXICO 2022.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice general

Índice de figuras	v
Resumen	XIII
1. Introducción	1
1.1. Objetivos y plan de trabajo	5
2. Cadenas poligonales	9
2.1. Definiciones básicas	9
2.2. Reconfiguración y encierro	13
2.2.1. Cadenas de ángulo interno fijo	22
2.3. Aplicación a la estructura proteínica	26
2.4. Teoría de nudos y funcionales de energía	35
2.5. Minimización de energía	39
3. Simulación	41
3.1. Descripción general	42
3.2. Intersección segmento-segmento rotante	42
3.2.1. Caso 1: ningún segmento está contenido en un plano paralelo al XY	43
3.2.2. Caso 2: el segmento que rota está en un plano paralelo al eje XY	51
3.2.3. Caso 3: Ambos segmentos están en un plano paralelo al eje XY	53
3.2.4. Decidibilidad de modificar una cadena rotando un ángu- lo diedro	57
3.3. Algoritmos para optimización	61
3.3.1. Recocido simulado	61

3.3.2. Algoritmos Genéticos	65
3.3.3. Descenso de gradiente fraccionario (FGD)	68
3.4. Otras energías posibles	73
4. Resultados	75
4.1. Mutaciones e hiperparámetros	76
4.2. Dicusión general	78
4.3. Transición de encierro	83
5. Conclusiones	93
Referencias	95
Anexo A: ejemplos de optimización	101
Anexo B: detalles sobre resultados	113

Índice de figuras

1.1.	Variedad de Calabi-Yau [3].	1
1.2.	Representación 3D de un fragmento de ADN [12].	3
1.3.	Cadena poligonal formada por $n = 10$ segmentos.	4
1.4.	Angulo diedro entre 4 puntos.	5
1.5.	Cadena poligonal encerrada.	6
2.1.	Angulo diedro ϕ entre 4 puntos.	12
2.2.	Restricciones para la intersección entre un radio (negro sólido) y un segmento que intersecta la esfera (rojo y azul punteado). El segmento de gran círculo (negro punteado) representa los puntos sobre la superficie tales que un radio de la esfera que pase por ese punto se intersectaría con dicho segmento de recta. Las posibles intersecciones solo se dan con el pedazo de segmento que está al interior de la esfera (rojo punteado) . . .	16
2.3.	Cadena poligonal encerrada con $n = 5$. El punto rojo es el punto medio de la curva que recorre los vértices v_1, \dots, v_4 . Inspirada en [14].	19
2.4.	Movimiento que intenta llevar la cadena de la figura 2.3 a una configuración plana.	19
2.5.	Cadena P con proyección sobre un plano P' sin intersección. Inspirada en [14].	20
2.6.	Paso inicial en la prueba del teorema 2.8. (a) \rightarrow (b) primero rotamos \mathbf{v}_0 hasta estar en el círculo verde de radio δ (b) \rightarrow (c) rotamos \mathbf{v}_0 sobre el eje del cilindro hasta estar en el plano Π_2	21
2.7.	Paso intermedio en la prueba del teorema 2.8. Notemos como, al movernos de un cilindro a otro, la cadena siempre queda contenida dentro de dicho cilindro.	21

2.8.	Cadena encerrada para ángulo interno fijo.	23
2.9.	(a) Cadena poligonal no aplanable con $n = 4$. Explícitamente, los vértices son $\mathbf{v}_0 = (0, -1/2, \epsilon)$, $\mathbf{v}_1 = (0, 1, 0)$, $\mathbf{v}_2 = (0, 0, 0)$, $\mathbf{v}_3 = (1, 0, 0)$, $\mathbf{v}_4 = (-\epsilon, 0, 3\epsilon/2)$. Los círculos rojo y naranja muestran el posible movimiento de los vértices \mathbf{v}_0 y \mathbf{v}_4 (b) movimiento necesario para aplanar la cadena.	24
2.10.	Primera parte de la cadena usada en el teorema 2.14.	25
2.11.	Segunda parte de la cadena usada en el teorema teorema 2.14. El último enlace tiene longitud 3.	25
2.12.	Cadena completa para el teorema 2.14.	26
2.13.	Movimiento para aplanar la cadena de la figura 2.11 (a) se colocan los enlaces de la partición en el rectángulo del cerrojo. (b) Se rota para colocar el enlace rojo en el triángulo azul. . .	27
2.14.	(a) Diagrama esquemático de un aminoácido. (b,c) Dos aminoácidos, leucina e isoleucina, con sus respectivas cadenas laterales.	27
2.15.	Reacción de condensación en la que se genera un enlace péptido entre dos aminoácidos.	28
2.16.	(a) Esquema de una proteína. La cadena principal se muestra punteada en color naranja. (b) Distancias características de los enlaces en una proteína [30, 31, 14].	29
2.17.	Estructura 3D de la proteína 3TJW [32]. La cadena principal se muestra en color naranja.	30
2.18.	(a) Estructura 3D de un ribosoma, tomada de [34] (b,c,d) Modelo simple de un ribosoma cónico produciendo una cadena poligonal [33].	31
2.19.	Distintas representaciones de algunos nudos. (a,b) pertenecen a la misma clase de equivalencia mientras que (c,d,f) pertenecen a otra.	37
3.1.	Superficies que contienen a las descritas por la rotación de un segmento de recta en \mathbb{R}^3 : hiperboloide de un manto, cono y cilindro.	47
3.2.	Regiones descritas por la rotación de un segmento de recta en \mathbb{R}^3 por un ángulo menor a 2π . Las regiones están contenidas en un hiperboloide y en un cono, respectivamente.	48
3.3.	Distintas configuraciones para la región generada al rotar una recta contenida en un plano paralelo al XY . En (a) los puntos sobre el segmento más cerca y lejos del origen del origen son los extremos, mientras que en (b) el punto más cercano no corresponde a ningún extremo.	52

3.4.	Posibilidades de la intersección respecto a la frontera entre un segmento de recta y la región descrita por otro segmento que rota en el mismo plano. (a) El segmento está totalmente contenido en la región. (b) Hay un pedazo contenido y otro fuera.	55
3.5.	Subcasos de posibles intersecciones para el caso 3 cuando el segmento no está completamente contenido en la región. . . .	58
3.6.	Probabilidad de aceptación de un cambio de configuración como función de $\Delta f = f(x') - f(x)$ para distintas temperaturas.	63
3.7.	Distintos programas de temperatura. La escala del eje x es logarítmica.	64
3.8.	Métodos de selección para un algoritmo genético: truncamiento con $k = 4$ (arriba), torneo con $k = 4$ (enmedio) y de ruleta(abajo).En la primera columna, una gráfica de barras con los valores de una función f para una población de tamaño $m = 8$. En la segunda columna, para el truncamiento se muestra en color la muestra aleatoria, para el torneo los mejores k elementos y para la ruleta, los valores de la función transformados en probabilidad. La última columna tiene al individuo seleccionado como padre.	67
3.9.	Comparación entre el flujo descrito por el gradiente convencional (rojo) y el gradiente fraccionario para la energía de la ecuación 2.18 con $p = 2$, $q = 4.5$ (azul). La longitud de los vectores es proporcional a su magnitud.	70
4.1.	Cadenas que buscamos llevar a otra configuración con el proceso de minimización.	75
4.2.	Aproximación de una curva a través de movimientos coordinados. La aproximación naranja tiene menor movimiento y es mejor a la curva.	77
4.3.	Últimas configuraciones para simulaciones con algoritmo genético sobre la cadena (a) agujas con $l = 2$ y (b) rectangular con $l = 1.15$	80
4.4.	Últimas configuraciones para simulaciones con algoritmo genético sobre la cadena (a) agujas con $l = 2$ y (b) rectangular con $l = 1.15$	81
4.5.	Cadena agujas en alta resolución.	82

4.6.	Resultados de simulación como función de l para la energía de punto tangente con el algoritmo Annealing. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones. . . .	84
4.7.	Resultados de simulación como función de l para la energía de punto tangente con el algoritmo genético. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.	85
4.8.	(a) Energía tangente promedio y (b) máximo del MaxSpan como función de l para la cadena rectangular con algoritmo Annealing.	86
4.9.	Resultados de simulación como función de l para la energía de MaxSpan con el algoritmo Annealing. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.	87
4.10.	Resultados de simulación como función de l para la energía de MaxSpan con el algoritmo genético. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.	88
4.11.	(a) SpecialSpan promedio y (b) máximo del SpecialSpan como función de l para la cadena rectangular con algoritmo Annealing con $l \in [1, 1.2]$	89
4.12.	(a) SpecialSpan promedio y (b) máximo del SpecialSpan como función de l para la cadena rectangular con algoritmo Annealing con $l \in [1, 1.4]$	90
4.13.	(a) MaxSpan promedio y (b) máximo del MaxSpan como función de l para la cadena agujas con algoritmo Annealing. . . .	91
A.1.	Gráfica de la función Eggholder en \mathbb{R}^3 y como mapa de calor.	103
A.2.	Minimización de la función Eggholder mediante recocido simulado. Una animación del proceso puede encontrarse en https://youtu.be/20TV63bm3Gk	104
A.3.	Minimización de la función Eggholder mediante un algoritmo genético. Una animación del proceso puede encontrarse en https://youtu.be/OvmFI_bhOWI	105
A.4.	Gráfica de la función de Mishra en \mathbb{R}^3 y como mapa de calor	106
A.5.	Minimización de la función de Mishra mediante recocido simulado. Una animación del proceso puede encontrarse en https://youtu.be/3YB_0BrOwWQ	107

A.6. Minimización de la función Mishra mediante un algoritmo genético. Una animación del proceso puede encontrarse en https://youtu.be/s72096Y7nwQ	108
A.7. Gráfica de la función de Gomez-Levy en \mathbb{R}^3 y como mapa de calor	109
A.8. Minimización de la función de Gomez-Levy mediante recocido simulado. Una animación del proceso puede encontrarse en https://youtu.be/4Dy0Ww7t-g	110
A.9. Minimización de la función de Gomez-Levy mediante un algoritmo genético. Una animación del proceso puede encontrarse en https://youtu.be/0uoD-S3dmVo	111
B.1. Trayectorias de minimización mediante annealing para la cadena aguja para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) -MaxSpan, y d) TP	117
B.2. Trayectorias de minimización mediante genético para la cadena aguja para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) -MaxSpan, y d) TP	118
B.3. Trayectorias de minimización mediante annealing para la cadena rectangular para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) -MaxSpan, d) -SpecialMaxSpan y e) TP	120
B.4. Trayectorias de minimización mediante algoritmo genético para la cadena rectangular para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) -MaxSpan, d) -SpecialMaxSpan y e) TP	122
B.5. Simulación de la cadena aguja, minimizando la energía del MaxSpan (ecuación 3.42) mediante Annealing.	125
B.6. Simulación de la cadena aguja, minimizando la energía de punto tangente (ecuación 3.32) mediante el algoritmo FGD.	126
B.7. Simulación de la cadena aguja, minimizando la energía del MaxSpan (ecuación 3.42) mediante el algoritmo genético.	127
B.8. Simulación de la aguja rectangular con $l = 2.589$, minimizando la energía del MaxSpan (ecuación 3.42) mediante el algoritmo Annealing con hiperparámetros modificados. Se tomó una temperatura inicial de 15 (en unidades de la energía inicial) y una final de 10^5 , con un número máximo de 300,000 iteraciones.	128
B.9. Simulación de la cadena rectangular, minimizando la energía del SpecialSpan (ecuación 3.43) mediante el algoritmo genético.	129

B.10. Simulación de la cadena rectangular, minimizando la energía del SpecialSpan (ecuación 3.32) mediante el algoritmo genético.	130
B.11. Simulación de la cadena rectangular con $l = 1.319$, minimizando la energía del SpecialSpan (ecuación 3.43) mediante el algoritmo Annealing con hiperparámetros modificados. Se tomó una temperatura inicial de 15 (en unidades de la energía inicial) y una final de 10^5 , con un número máximo de 300,000 iteraciones.	131
B.12. Simulación del nudo figura 8 en versión polinomial, minimizando la energía de punto tangente mediante el algoritmo FGD.	132
B.13. Simulación del nudo trefoil en versión polinomial, minimizando la energía de punto tangente (ecuación 3.32) mediante el algoritmo FGD.	133

Agradecimientos

Desde un lado personal, quisiera primero agradecer a mi mamá por todo el apoyo que me ha brindado a lo largo de mi vida.

A Sofía Giordano Vélez, que me acompañó por sesiones de trabajo en la madrugada, comidas de 15 minutos y clases en línea, siempre con el mayor de los cariños.

A todos mis amixes. En particular, a Andres Mejía Zacarías, por sus palabras de aliento y agradables conversaciones, y a Jesus Fernando Ramírez Barrios, que me ayudó a corregir la redacción de este trabajo.

A todxs quienes fueron alumnxs de los 6 cursos que impartí durante la maestría. Sus dudas, comentarios y curiosidades siempre me enriquecieron.

En el lado profesional, primero agradezco a mi asesor, Atahualpa S. Kraemer, no solo por su guía, apoyo y amistad a lo largo de más de 5 años si no por mostrarme todo lo positivo que puede surgir de una relación académica sana.

A los profesores David P. Sanders y Pedro Miramontes Vidal, por darse un espacio para aconsejarme y resolver mis dudas existenciales.

A mis sinodales por sus valiosos comentarios y el tiempo dedicado a este trabajo.

A toda la coordinación del Posgrado en Ciencias Matemáticas. En particular a Lucía y a Tere por su paciencia para ayudarme con la burocracia.

Este trabajo se realizó con apoyo de una beca nacional para estudios de maestría de CONACYT. También hubo apoyo por parte del Laboratorio de Cómputo de Alto Rendimiento de la Facultad de Ciencias de la UNAM.

Resumen

Una cadena poligonal con restricciones sobre la longitud de sus enlaces y el valor de sus ángulos internos puede servir como modelo de la cadena principal de una proteína. Existen casos en los cuales el espacio de configuraciones de dichas cadenas no es conexo, condición conocida como encierro. Es posible relacionar la existencia de cadenas encerradas con la paradoja de Levinthal mediante un modelo probabilístico de la síntesis de una proteína.

Inspirados en la minimización de energía utilizada en teoría de nudos, en este trabajo construimos una metodología numérica para intentar decidir si una cadena poligonal está encerrada. Mediante un proceso de minimización, intentamos llevar a dicha cadena hacia una configuración canónica que minimiza una función de energía dada. Utilizamos algoritmos metaheurísticos, que nos permiten modificar la cadena sin salirnos de su componente conexas en el espacio de configuraciones, así como algoritmos basados en descenso de gradiente que han mostrado resultados exitosos en problemas de desenredamiento de curvas suaves.

Abstract

A polygonal chain with constraints over the length of its links and the value of its bond angles can be used as a model of a protein backbone. There are known cases where the configuration spaces of such chains is disconnected, a condition known as locking. It is possible to relate the existence of locked chains to the Levinthal paradox using a probabilistic model of protein synthesis.

Inspired by energy minimization approaches in knot theory, in this work a numerical methodology is built to try to decide if a polygonal chain is locked. We try to move a chain to a canonical configuration that minimizes an energy function. We use metaheuristic algorithms, which allow us to modify chains without creating invalid configurations, and algorithms based on gradient descent, which have shown promising results untying smooth knots.

Capítulo 1

Introducción

¡Oh rey! para viajar por tu imperio hay calzadas para la realeza y sendas para caminantes, pero en geometría hay un solo camino para todos

Respuesta de Menecmo a Alejandro Magno, según Estobeo [1, 2]

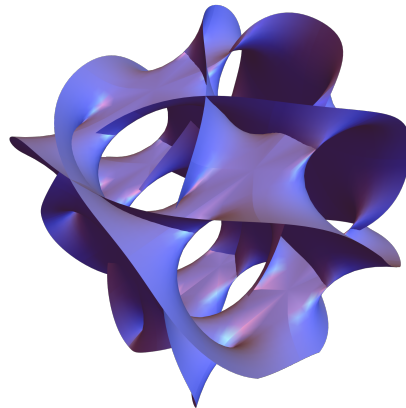


Figura 1.1: Variedad de Calabi-Yau [3].

El significado puramente etimológico de la palabra “geometría” se puede entender como “medición de la tierra”. Por otro lado, una definición más

contemporánea describe a la geometría como “la ciencia del espacio” [4]. Dicha definición puede construirse sobre el trabajo filosófico de Kant, según el cual el espacio es una intuición pura que sirve de base a todas las intuiciones externas [5], por lo que es el lugar en el que se desarrolla toda la geometría. Es posible argumentar que ciertas propiedades destacadas de esta rama de las matemáticas como contar con el sistema axiomático más antiguo registrado¹ [1] o la importancia que tiene en la descripción física de la naturaleza [7] apoyan la idea de que el espacio es una intuición pura que precede a la experiencia sensorial.

Una de las propiedades más importantes de la geometría es su capacidad para plantear problemas que son sencillos de entender de forma intuitiva, pero difíciles de formular en un lenguaje matemático abstracto y todavía más complicados de demostrar de forma rigurosa. Dos ejemplos canónicos de este tipo de problemas son el teorema de los cuatro colores [8] y la conjetura de Kepler [9]. Cabe mencionar que ambos necesitaron de ayuda computacional mediante verificadores formales y asistentes de pruebas para concretar una prueba formal.

Al modelar y simular procesos físicos, químicos y biológicos en una computadora, la geometría, en particular la geometría discreta, juega un papel fundamental en la descripción correcta de dichos procesos. Algunos ejemplos son la detección de colisiones en simulaciones de partículas [10] y el mallado para análisis de elemento finito [11]. En el caso específico de la bioquímica molecular, las estructuras geométricas descritas por macromoléculas como proteínas, ácidos nucleicos y virus llegan a ser sumamente complejas, como muestra la figura 1.2.

¹Que además sirvió como inspiración para el proyecto formalista de Hilbert a principios del siglo XX [6]

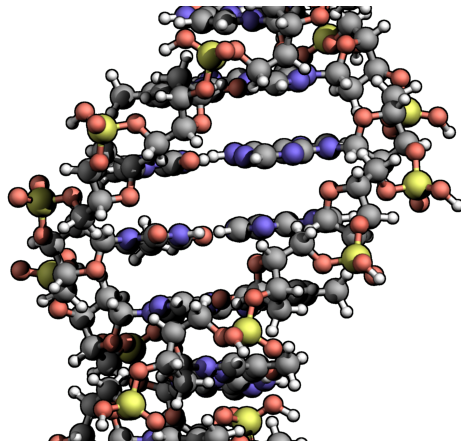


Figura 1.2: Representación 3D de un fragmento de ADN [12].

Particularmente, la estructura tridimensional de una proteína determina la función biológica que realiza. Más aún, dentro de su composición química está codificada la estructura más estable que deben de adquirir para cumplir su función biológica, llamada **estructura nativa**. Debe mencionarse que las proteínas no se sintetizan en su estructura nativa, si no en otra y se deforman continuamente hasta alcanzarla. Dicho proceso se conoce como **plegamiento de proteínas**.

Si una proteína no logra plegarse hasta su estructura nativa, esta no cumplirá su función biológica y generará problemas como enfermedades autoinmunes. Por ello, existe un gran interés en describir el proceso mediante el cual una proteína se transforma hasta llegar a su estructura nativa. Este problema se vuelve más complicado al considerar que la cantidad de estructuras que puede adoptar una proteína es enorme² para el tiempo que le toma adquirir dicha estructura³. Este hecho aparentemente paradójico es conocido como la **paradoja de Levinthal** [13].

Es posible que este número de estructuras sea menor al especulado: puede haber restricciones de naturaleza geométrica sobre las estructuras accesibles desde cierta configuración. Si esto fuera cierto, explicaría parcialmente la discrepancia entre los tiempos de plegamiento y la cantidad de configuraciones accesibles. Para analizar dicha suposición, es necesario hacer una

²Se estiman $\sim 10^n$ configuraciones para una proteína con n aminoácidos

³Entre 10^{-6} – 10^2 segundos, dependiendo del número de aminoácidos y otras propiedades

descripción geométrica simple de una proteína.

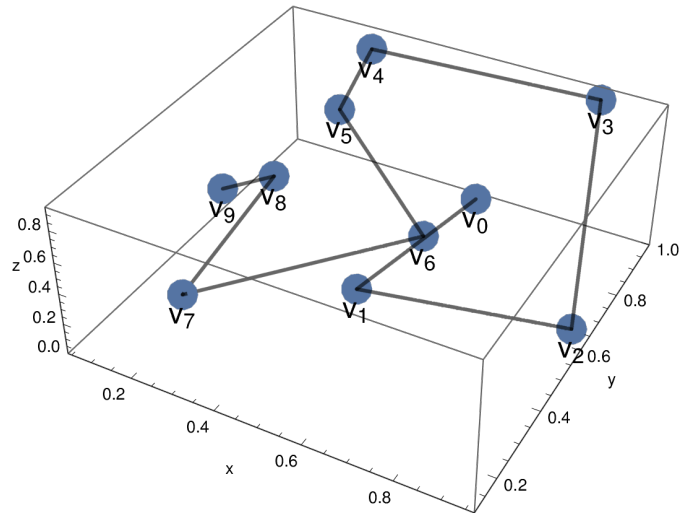


Figura 1.3: Cadena poligonal formada por $n = 10$ segmentos.

La descripción más básica de la estructura de una proteína se define a través de su cadena principal, la cual se puede modelar como una cadena poligonal. Una cadena poligonal es una curva lineal a pedazos en \mathbb{R}^3 . Es decir, es una curva compuesta por n segmentos de recta, como muestra la figura 1.3. Es posible especificar dicha curva con $n + 1$ vértices en \mathbb{R}^3 , y, quitando traslaciones y rotaciones, describir la cadena con $3n - 3$ grados de libertad. Dichos grados consisten en las n longitudes de sus segmentos, los $n - 1$ ángulos internos entre segmentos sucesivos y los $n - 2$ ángulos diedros, que son ángulos entre los planos generados por cuatro vértices sucesivos, como muestra la figura 1.4.

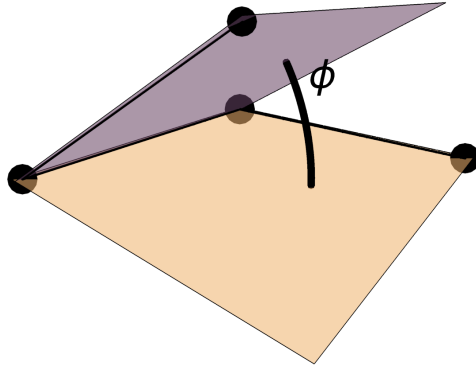


Figura 1.4: Ángulo diedro entre 4 puntos.

Debido a propiedades de su estructura, la cadena principal de una proteína corresponde a una cadena poligonal en la que las longitudes de los segmentos y los ángulos internos están fijos, y los únicos grados de libertad están dados por sus $n - 2$ ángulos diedros [14].

Nos gustaría poder decidir, dada una cadena poligonal, si su espacio de configuraciones es conexo: es decir, si es posible transformar una cadena modificando sus grados de libertad (sus ángulos diedros, sus ángulos diedros e internos, etc) para llevarla a otra configuración. Existen diversos casos, cuando la cadena tiene “nudos”, en los que sabemos que no es posible. La figura muestra uno de estos casos. Cuando la cadena tiene un espacio de configuraciones desconexo, decimos que la cadena está **encerrada**. Actualmente, no existe un algoritmo que permita decidir si una cadena está encerrada [14, 15].

1.1. Objetivos y plan de trabajo

En este trabajo, buscamos atacar dicho problema mediante un proceso de minimización: al definir una función (que llamaremos **energía**) en el espacio de configuraciones de una cadena, buscamos ver si desde cualquier configuración es posible ir a aquella que minimiza globalmente la energía. Dependiendo de la función de energía, la configuración que minimiza dicha función puede tener distintas propiedades deseadas, como estar contenida en un plano o en una recta. En la teoría de nudos, el mismo enfoque de minimización es utilizado por algunos autores para decidir si un nudo es equivalente a otro [16, 17, 18].

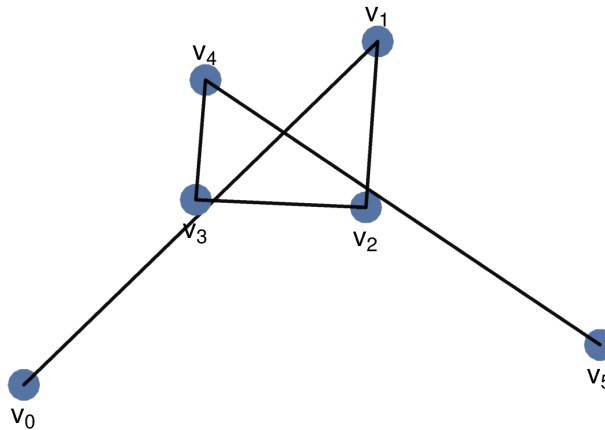


Figura 1.5: Cadena poligonal encerrada.

Para el proceso de minimización, utilizamos dos tipos de algoritmos: metaheurísticas, como el recocido simulado (*simulated annealing*) [19, 20] y los algoritmos genéticos [21], y algoritmos basados en descenso de gradiente [18, 22]. Los primeros tienen la ventaja de permitirnos verificar en cada paso si los movimientos que realizamos para modificar una configuración son válidos, es decir, si no generan intersecciones, mientras que los últimos dependen de propiedades de la función a minimizar y de la tasa de movimiento para no generar intersecciones. Estrictamente, no hay garantía de que los algoritmos basados en descenso de gradiente no generen intersecciones.

Para verificar que un cambio de configuración se puede realizar sin generar una intersección entre segmentos de la cadena, debemos verificar si la superficie descrita por los segmentos que rotan al cambiar un ángulo de la cadena (una región finita contenida en un hiperboloide de un manto) interseca a los segmentos que se quedarían estáticos. Es necesario desarrollar la teoría, mediante geometría analítica elemental, de cómo detectar dicha intersección.

Condensando todo lo anterior, nuestro objetivo es simular de forma numérica el llevar una cadena poligonal desde una configuración dada hasta una configuración canónica como un proceso de minimización. Para eso, desarrollaremos un algoritmo que nos diga si es posible cambiar un ángulo diedro o interno de una cadena sin generar intersecciones. Posteriormente, utilizaremos dicho algoritmo para generar nuevas configuraciones de una ca-

dena a partir de otra y así poder emplear algoritmos de optimización, tanto metaheurísticas como algoritmos basados en descenso de gradiente, para minimizar algunas funciones sobre la cadena que la lleven a la configuración canónica deseada.

Este trabajo no pretende resolver de forma estricta las proposiciones geométricas sobre el encierro de ciertas cadenas poligonales, sino simplemente hacer un análisis numérico del problema. Se procede de la siguiente manera: el capítulo 2 define de forma rigurosa las cadenas poligonales y la teoría asociada a ellas, así como algunas propiedades básicas de su espacio de configuraciones, qué es una proteína y por qué se puede modelar con una cadena poligonal y la relación entre la paradoja de Levinthal y las cadenas poligonales encerradas.

En el capítulo 3 desarrollamos la teoría necesaria sobre las simulaciones numéricas: el algoritmo para detectar si es posible cambiar un grado de libertad de la cadena, que se vuelve una búsqueda exhaustiva de intersecciones entre segmentos que rotan y otros estáticos. Así, ahí desarrollamos exponeremos la teoría para determinar si se da la una intersección entre la región delimitada por un segmento rotante y uno estático. También expondremos la teoría sobre el recocido simulado, los algoritmos genéticos y el descenso de gradiente fraccionario.

En el capítulo 4 mostramos los resultados de las simulaciones para casos particulares de cadenas poligonales, así como para cadenas parametrizadas por una longitud buscando si es perceptible una transición entre una cadena encerrada y una libre como función de dicho parámetro.

Finalmente, en el capítulo 5 damos algunas conclusiones sobre lo observado en las simulaciones y sugerimos algunas acciones para realizar trabajo a futuro.

Capítulo 2

Cadenas poligonales y transición de encierro

En esta sección, daremos las definiciones matemáticas básicas sobre las cadenas poligonales. Posteriormente, expondremos cómo se puede aplicar dicha teoría para modelar la cadena principal de una proteína y mostraremos cómo se relacionan estas áreas con la teoría de nudos, la rama de la topología que se encarga de estudiar y clasificar los nudos en tres dimensiones.

2.1. Definiciones básicas

Para dar las definiciones, seguimos el tratamiento de [14] haciendo uso también de [15]. A falta de referencias bibliográficas en español, los términos usados para denotar algunos objetos se emplean por primera vez. Pondremos el término en inglés para evitar confusiones o problemas.

Definición 2.1. Una **articulación** (*linkage* en inglés) es un conjunto S de segmentos de recta tales que dado un segmento arbitrario $s_i = \{v_{i_1}, v_{i_2}\} \in S$, con v_i un vértice abstracto que representa un extremo del segmento, existe al menos otro segmento $s_j = \{v_{j_1}, v_{j_2}\} \in S$ tal que $s_i \cap s_j \neq \emptyset$, es decir, cualquier segmento de una articulación tiene un extremo que también es el extremo de otro segmento. A los segmentos de una articulación también se les llama **enlaces** (*link*).

Notemos que la definición de enlace es general: no hay necesidad de

especificar en qué espacio se encuentran los segmentos siempre y cuando sean rectas. Inclusive es posible generalizar dicha definición tomando curvas geodésicas en un espacio no euclideo.

A partir de esta definición, existen diversas formas de categorizar articulaciones. La primer forma es por el espacio en el que viven los segmentos de la articulación. En general, la mayoría de la investigación está relacionada con articulaciones en espacios euclidianos \mathbb{R}^d . Es posible construir una gráfica a partir de una articulación, tomando como vértices los extremos de los segmentos y como aristas los segmentos mismos. Así, otra forma de categorizar a una articulación es según las propiedades topológicas de la gráfica asociada a ella.

Para diversas aplicaciones, en robótica y biología, es importante considerar el caso en el cual habrá obstáculos en el espacio con los que la articulación no puede intersectarse. En particular, es muy relevante el caso en cual los segmentos de la articulación no pueden intersectarse entre sí en ningún otro punto que no sea un extremo. Esto provee otra forma de categorizar a las articulaciones según las restricciones espaciales que pueda tener.

Categoría	Clases
Espacio	$\mathbb{R}^2, \mathbb{R}^3, S^2$
Gráfica asociada	General, Árbol, Trayectoria, Polígono
Restricciones espaciales	Ninguna, Sin intersecciones, Obstáculos

Tabla 2.1: Clases para una cadena poligonal.

La tabla 2.1 resume todas las formas mencionadas de categorizar una articulación. Es importante notar que el tipo de espacio donde exista la articulación también define la una dimensión en la que existe la articulación. Nosotros solo estamos interesados en articulaciones en espacios euclidianos, por lo que cuando nos refiramos a una articulación de dimensión 2 nos referimos a una articulación que vive en \mathbb{R}^2 .

En este trabajo nos interesa en particular un tipo particular de articulaciones.

Definición 2.2. Una **cadena poligonal** (*polygonal chain*) de longitud n es una articulación de n segmentos cuya gráfica asociada es una trayectoria de longitud n y que tiene la restricción espacial de no permitir intersecciones.

Aunque una cadena poligonal, al igual que una articulación, es un objeto abstracto, en general vamos a trabajar con realizaciones geométricas concretas de dicho objeto. Es sencillo notar que una cadena poligonal está únicamente definida por los vértices de los extremos de sus segmentos. Así, podemos denotar una cadena poligonal de dimensión d como un vector de $n + 1$ vértices $P = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n)$ con $\mathbf{v}_n \in \mathbb{R}^d$. Durante este trabajo, trataremos con cadenas poligonales de dimensión 3, por lo que a partir de este momento se obviará que cada que hagamos mención de una cadena poligonal esta existe en \mathbb{R}^3 .

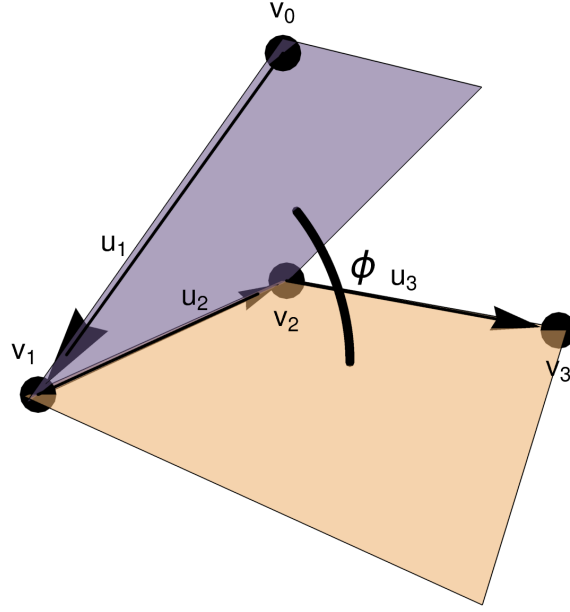
En general, una cadena poligonal de longitud n tiene $3n + 3$ grados de libertad que se relacionan con las 3 coordenadas de sus $n + 1$ vértices. Sin embargo, 6 de estos grados de libertad surgen de las traslaciones y rotaciones de la cadena. No nos interesa tomar en cuenta estos grados de libertad pues no queremos distinguir entre dos cadenas poligonales que se puedan empalmar completamente tras una traslación y rotaciones adecuadas. Así, tenemos en total $3n - 3$ grados de libertad en una cadena.

Además de la representación explícita en coordenadas cartesianas de cada vértice, es posible describir una cadena a partir de sus coordenadas internas. Antes de explicar cuales son las coordenadas internas de una cadena, necesitamos una definición geométrica.

Definición 2.3. Dados cuatro puntos $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^3$, el **ángulo diedro** (*dihedral angle*) ϕ corresponde al ángulo entre dos planos: el plano generado por el vector $\mathbf{v}_1 - \mathbf{v}_0$ y el vector $\mathbf{v}_2 - \mathbf{v}_1$ y el generado por el vector $\mathbf{v}_3 - \mathbf{v}_2$, como se muestra en la figura 2.1.

Notemos que, gracias al orden de los puntos, es posible garantizar que el ángulo diedro estará en el intervalo $[0, 2\pi)$ pues podemos usar el vector $\mathbf{v}_2 - \mathbf{v}_1$, coincidente entre ambos planos, como la dirección de la rotación entre las normales a los planos.

Dado $\mathbf{u}_i = \mathbf{v}_i - \mathbf{v}_{i-1}$ el vector del enlace entre el vértice \mathbf{v}_i y su subsecuente, las coordenadas internas consisten en las n longitudes de cada enlace $l_i = \|\mathbf{u}_i\|$, los $n - 1$ ángulos internos entre parejas de enlaces θ_i definidos

Figura 2.1: Ángulo diedro ϕ entre 4 puntos.

como¹

$$\theta_i = \arctan 2 (\|\mathbf{u}_i \times \mathbf{u}_{i+1}\|, \mathbf{u}_i \cdot \mathbf{u}_{i+1}), \quad (2.1)$$

y los $n - 2$ ángulos diedros ϕ_i , calculados como

$$\phi_i = \arctan 2 (\mathbf{u}_{i+1} \cdot (\mathbf{n}_i \times \mathbf{n}_{i+1}), \|\mathbf{u}_{i+1}\|(\mathbf{n}_i \cdot \mathbf{n}_{i+1})), \quad (2.2)$$

con $\mathbf{n}_i = \mathbf{u}_i \times \mathbf{u}_{i+1}$. La ecuación 2.2 se puede simplificar al usar la identidad $(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = ((\mathbf{a} \times \mathbf{b}) \cdot \mathbf{d})\mathbf{c} - ((\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c})\mathbf{d}$, por lo que

$$\begin{aligned} \mathbf{n}_i \times \mathbf{n}_{i+1} &= (\mathbf{u}_i \times \mathbf{u}_{i+1}) \times (\mathbf{u}_{i+1} \times \mathbf{u}_{i+2}) \\ &= (\mathbf{u}_i \times \mathbf{u}_{i+1}) \times (\mathbf{u}_{i+1} \times \mathbf{u}_{i+2}) \\ &= ((\mathbf{u}_i \times \mathbf{u}_{i+1}) \cdot \mathbf{u}_{i+2})\mathbf{u}_{i+1} - ((\mathbf{u}_i \times \mathbf{u}_{i+1}) \cdot \mathbf{u}_{i+1})\mathbf{u}_{i+2} \\ &= ((\mathbf{u}_i \times \mathbf{u}_{i+1}) \cdot \mathbf{u}_{i+2})\mathbf{u}_{i+1}. \end{aligned} \quad (2.3)$$

El paso del último renglón se dio ya que $\mathbf{u}_i \times \mathbf{u}_{i+1} \perp \mathbf{u}_{i+1}$. Así, la ecuación 2.2 se convierte en

$$\begin{aligned} \phi_i &= \arctan 2 (\mathbf{u}_{i+1} \cdot ((\mathbf{u}_i \times \mathbf{u}_{i+1}) \cdot \mathbf{u}_{i+2})\mathbf{u}_{i+1}, \|\mathbf{u}_{i+1}\|(\mathbf{n}_i \cdot \mathbf{n}_{i+1})) \\ &= \arctan 2 (\|\mathbf{u}_{i+1}\|((\mathbf{u}_i \times \mathbf{u}_{i+1}) \cdot \mathbf{u}_{i+2}), (\mathbf{n}_i \cdot \mathbf{n}_{i+1})). \end{aligned} \quad (2.4)$$

¹La función $\arctan 2(y, x)$ es la extensión de \arctan que corrige las diferencias de signos para que su imagen sea todo el intervalo $[0, 2\pi)$ en lugar de solo $[0, \pi)$.

En general, trataremos de representar una cadena mediante sus coordenadas internas por ser más útiles para explorar las propiedades que nos interesan en este trabajo. Sin pérdida de generalidad, denotaremos con Ω el espacio de todos los posibles valores de las coordenadas internas de una cadena, es decir, todas sus posibles configuraciones.

Cuando estudiamos articulaciones con la restricción de que no se interseccionen los segmentos, visualizamos a los segmentos como barras rígidas. Por lo tanto, generalmente las longitudes de los enlaces l_i de una cadena poligonal se consideran constantes y no un grado de libertad del sistema, mientras que los ángulos internos y los diedros conforman $2n - 3$ grados de libertad del sistema. También es importante notar es que una cadena poligonal puede ser quiral ².

2.2. Reconfiguración y encierro

Existen diversas preguntas que nos podemos hacer sobre las cadenas poligonales. Sin embargo, los más relevantes están relacionadas con cómo es que la geometría de la cadena y sus restricciones afectan la posibilidad de ir de una configuración de la cadena a otra, es decir, de reconfigurar la cadena con transformaciones continuas. Destacamos tres problemas importantes:

1. **Alcanzabilidad** (*Reachability*): dado un punto $\mathbf{v}^* \in \mathbb{R}^3$ contenido en una cadena P , ¿es posible hacer una transformación continua sobre la cadena para que el punto \mathbf{v}^* coincida con otro punto arbitrario arbitrario $\mathbf{a} \in \mathbb{R}^3$? (Típicamente \mathbf{v}^* es uno de los extremos \mathbf{v}_i , generalmente el n -ésimo).
2. **Reconfiguración** (*Reconfiguration*): Dadas dos configuraciones distintas $\omega_1, \omega_2 \in \Omega$ de una cadena, ¿es posible llevar la cadena de ω_1 a ω_2 mediante una transformación continua en el espacio de configuraciones?
3. **Encierro** (*Locking*) ¿Es conexo por trayectorias el espacio de configuraciones? (I.E. para todo para de configuraciones $\omega_1, \omega_2 \in \Omega$, la cadena se puede reconfigurar).

²Un conjunto $S \subset \mathbb{R}^3$ es quiral sí y sólo si no es posible empalmarlo totalmente sobre el conjunto obtenido al aplicarle una reflexión S' aplicando solamente rotaciones y traslaciones. La quiralidad es una propiedad importante en química pues normalmente las moléculas quirales tienen distintos comportamientos físicoquímicos dependiendo de su orientación.

Es claro que el problema de encierro es una versión más fuerte del problema de reconfiguración. Por su importancia, usaremos la siguiente definición.

Definición 2.4. Una cadena poligonal P está **encerrada** (*locked*) sí y sólo sí su espacio de configuraciones tiene más de una componente conexa.

Es posible responder a cada uno de estos problemas de distintas formas. El primer tipo de respuesta es simplemente contestar de forma tajante si la transformación deseada es posible o no. El segundo tipo de respuesta es no solo decir si es posible o no sino dar de forma explícita la transformación continua que resuelva el problema.

La pregunta que nos interesa atacar en esta tesis es la reconfiguración y el encierro de cadenas poligonales en tres dimensiones. En otras dimensiones los resultados también son interesantes pero escapan al alcance de este trabajo ³. Presentaremos algunos de los resultados analíticos que se han encontrado sobre el tema, siguiendo un tratamiento parecido a [23]

Lema 2.1. *Dada una cadena de longitud n en la que los enlaces pueden tener longitud variable, hay una longitud l tal que para $l_n < l$ el espacio de configuraciones del vértice \mathbf{v}_n es $B_{l_n}(\mathbf{v}_{n-1}) \setminus \mathbf{u}_{n-1}$ con $B_r(\mathbf{x})$ la bola de radio r centrada en \mathbf{x} .*

Demostración. Sin pérdida de generalidad, supongamos que tomamos un sistema coordenado tal que

1. \mathbf{v}_{n-1} está en el origen.
2. El segmento \mathbf{u}_{n-1} es paralelo al eje z .
3. El segmento \mathbf{u}_{n-2} esta contenido en el plano XZ .

En dicho caso, notemos que el ángulo interno entre θ_{n-1} corresponde al ángulo polar y el ángulo diedro ϕ_{n-2} corresponde al ángulo azimutal. Supongamos que hacemos el valor de l_n suficientemente pequeño tal que el único enlace que intersecta a dicha bola es \mathbf{u}_{n-1} . En dicho caso, podemos mover el segmento \mathbf{u}_n por toda la bola sin intersectar a ningún enlace, y el único momento donde podría intersectar uno es cuando se empalma con \mathbf{u}_{n-1} . ■

³Las cadenas poligonales en 2D o 4D no se pueden encerrar

Lema 2.2. *Dada una esfera de radio r y un segmento de recta $s \subset \mathbb{R}^3$ que intersecta a la esfera, entonces la línea descrita sobre la esfera cuando un radio recorre todas sus intersecciones con el segmento s es un segmento de círculo máximo de la esfera. Más aún, dicho segmento tiene longitud menor a πr .*

Demostración. Antes que nada, recordemos que círculo máximo de la esfera es el círculo descrito por la intersección de un plano que pasa por el centro de la esfera y la superficie de la esfera. Tomamos casos sobre los distintos tipos de intersecciones posibles entre el segmento s y la esfera:

1. s intersecta tangencialmente a la esfera.

La sección descrita al recorrer un radio por la intersección es un único punto, el cual trivialmente es un segmento de círculo máximo.

2. s intersecta al interior de la esfera y es colineal a un radio.

Ánalogo al caso anterior.

3. s intersecta al interior de la esfera y no es colineal a un radio.

Notemos que, dado un radio r de la esfera, existe un único plano que contiene a dicho radio y al segmento s . Más aún, dicho plano pasa por el origen de la esfera, por lo que la intersección del plano con la superficie de la esfera es un círculo máximo.

Al mover el radio r de tal forma que recorra todas sus posibles intersecciones con s , el plano generado por ambos segmentos no cambia ya que el plano donde se intersectan es único. Así, sobre la superficie de la esfera el punto que se recorre también está contenido en este plano y, por ende, es un segmento de círculo máximo. La figura 2.2 muestra dichos gráficos.

Para ver que la longitud de dicho segmento es menor a πr , notemos que, al no ser colineal con un radio, la intersección de s con el interior de la esfera siempre está contenida en una media esfera, por lo que al recorrerla el segmento recorrerá una longitud menor a πr . ■

Lema 2.3. *Dada una cadena de longitud n en la que los enlaces tienen*

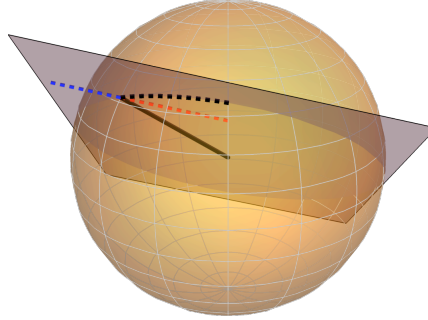


Figura 2.2: Restricciones para la intersección entre un radio (negro sólido) y un segmento que intersecta la esfera (rojo y azul punteado). El segmento de gran círculo (negro punteado) representa los puntos sobre la superficie tales que un radio de la esfera que pase por ese punto se intersectaría con dicho segmento de recta. Las posibles intersecciones solo se dan con el pedazo de segmento que está al interior de la esfera (rojo punteado)

longitudes fijas, el espacio de configuraciones del vértice \mathbf{v}_n es el cascarón esférico de radio l_n centrada en \mathbf{v}_{n-1}) menos el punto sobre la esfera que genera un radio colineal con el segmento \mathbf{u}_{n-1} y quitando también el conjunto $\{\hat{\mathbf{u}}_i \mid 1 < n - 1\}$, con $\hat{\mathbf{u}}_i$ el segmento de círculo máximo en el que se dan las intersecciones entre \mathbf{u}_n y \mathbf{u}_i .

Demostración. Análogamente a la prueba del lema 2.1, tomamos un mismo sistema de referencia y construimos la esfera centrada en \mathbf{v}_{n-1} . \mathbf{v}_n puede moverse en todos los lugares de la esfera que no provoquen que el segmento \mathbf{u}_n se intersecte con algún otro \mathbf{u}_i . Para el segmento \mathbf{u}_{n-1} , solo nos quita el punto de la esfera con un radio colineal al segmento \mathbf{u}_{n-1} . Para todos los otros segmentos \mathbf{u}_i , sabemos que las posiciones en las que no puede estar \mathbf{v}_n son aquellas que generan una intersección entre \mathbf{u}_i y \mathbf{u}_n . Por el lema 2.2, sabemos que dichas posiciones para el segmento \mathbf{v}_n estarán en un segmento de gran círculo. ■

Teorema 2.4. *Una cadena poligonal con enlaces de longitud variable nunca se encierra, es decir, es posible llevar una configuración a cualquier otra. Más aún, también es posible llevar cualquier vértice a una posición arbitraria.*

Demostración. La prueba formal se realiza por inducción sobre la longitud de la cadena. Es claro que una cadena de longitud 1 cumple todo lo men-

cionado. Suponiendo que una cadena de longitud n lo cumple, una cadena de longitud $n + 1$ tendrá las mismas propiedades gracias al lema 2.1 pues siempre es posible reducir su longitud para que podamos moverla en la dirección deseada, y entonces alargarla para hacer llegar el vértice \mathbf{v}_{n+1} a una posición deseada. ■

Teorema 2.5. *Ninguna cadena poligonal de longitud $n \leq 4$ con enlaces de longitud fija puede encerrarse.*

Demostración. Es claro que una cadena de longitud 1 y 2 nunca se encierra. Para ver que una cadena de longitud 3 o 4 nunca se encierra, usaremos un argumento recursivo similar a una inducción pero no para toda n . Así, supongamos que tenemos una cadena de longitud $n < 4$ que nunca se encierra y probemos que una cadena de longitud $n + 1 \leq 4$ tampoco se encierra

Por la hipótesis mencionada, la subcadena obtenida al remover el enlace \mathbf{u}_{n+1} nunca se encierra. Así, la única posibilidad de que la cadena se encierre es debido al movimiento del enlace \mathbf{u}_{n+1} . Por el lema 2.3, sabemos que el espacio de configuraciones del vértice \mathbf{v}_{n+1} es una esfera cuyas únicas restricciones por las posibles intersecciones se expresan como segmentos de grandes círculos.

Para que la cadena se encerrase, tendría que ser posible generar una región cerrada en la esfera con $n - 1 \leq 2$ segmentos de grandes círculos, cosa que claramente es imposible. Así, necesariamente la cadena no se puede encerrar. ■

El primer ejemplo de una cadena encerrada se puede dar para $n = 5$.

Lema 2.6. *Una cadena poligonal no está encerrada sí y sólo si para toda configuración $\omega \in \Omega$ es posible transformar continuamente la cadena hasta una configuración canónica arbitraria ω^* , como la que contiene a todos los puntos en una misma recta sin encimar sus enlaces.*

Teorema 2.7. *Sea P una cadena esquemática como la de la figura 2.3. Si $l_1, l_5 > l_2 + l_3 + l_4$ entonces la cadena está encerrada.*

Demostración. Antes de realizar la prueba, aclaramos que el segmento \mathbf{u}_1 pasa entre los segmentos \mathbf{u}_4 y \mathbf{u}_5 , mientras que análogamente \mathbf{u}_5 pasa entre

\mathbf{u}_1 y \mathbf{u}_2 , formando una estructura topológicamente ⁴ equivalente a un nudo trefoil con un corte.

Para demostrar que la cadena está encerrada usaremos el lema 2.6 y probaremos que no es posible llevarla a una configuración en la que todos los vértices estén contenidos en una recta. La esfera con radio $(l_1 + l_2 + l_3)/2$ y centrada en el punto medio de la curva que recorre los vértices v_1, \dots, v_4 , que se muestra en la figura 2.3, siempre contiene en su interior a los vértices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ y \mathbf{v}_4 . Sin embargo, por construcción, \mathbf{v}_0 y \mathbf{v}_5 siempre están fuera de la esfera. Sin pérdida de generalidad, notemos que para poder llevar a la cadena a una configuración contenida en una recta, sería necesario deshacer el nudo presente, lo cual necesita que los vértices \mathbf{v}_0 y \mathbf{v}_5 pasen por abajo de los ángulos entre $\mathbf{u}_5, \mathbf{u}_4$ y $\mathbf{u}_1, \mathbf{u}_2$, respectivamente, lo cual es imposible ya que estos vértices siempre están fuera de la esfera. La figura 2.4 muestra un intento de dicho movimiento.

Siendo más estrictos, podemos formalizar la idea anterior utilizando reducción al absurdo. Supongamos que existe un movimiento continuo que pone a todos los vértices como contenidos en la misma recta. Sin pérdida de generalidad, podríamos unir los vértices \mathbf{v}_0 y \mathbf{v}_5 con una curva suave y simple totalmente exterior a la esfera previo a dicho movimiento. Así, al final del movimiento, obtendríamos una estructura topológicamente equivalente a un círculo. Esto es una contradicción pues implicaría que el nudo trefoil es equivalente a un círculo. ■

Nos gustaría dar condiciones más generales que garantizaran que una cadena está o no encerrada. Existe un caso sencillo en cuál podemos dar dichas condiciones.

Teorema 2.8. *Sea P una cadena poligonal tal que existe un plano en el cual la proyección 2D de la cadena, denotada P' , no se intersecta como en la figura 2.5. Entonces la cadena no está encerrada.*

Demostración. Nuevamente, utilizaremos el lema 2.6. Sin pérdida de generalidad, suponemos que el plano sobre el cual la cadena no se intersecta es el plano XY . Dada la cadena en dos dimensiones $P' = (\mathbf{v}'_0, \dots, \mathbf{v}'_n)$ Sea ahora

⁴Con la ligera diferencia de no ser una curva suave

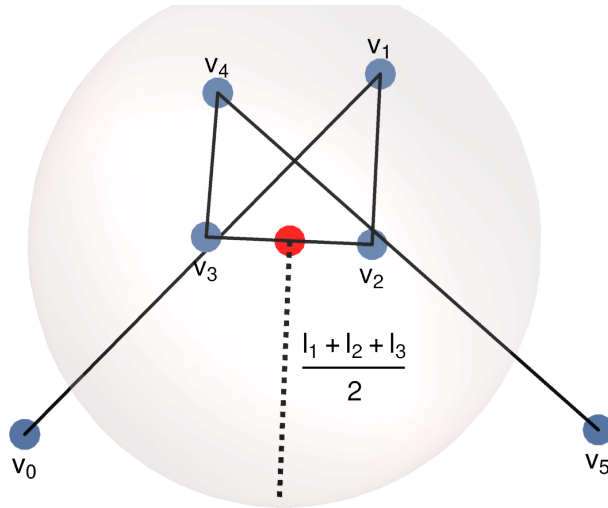


Figura 2.3: Cadena poligonal encerrada con $n = 5$. El punto rojo es el punto medio de la curva que recorre los vértices v_1, \dots, v_4 . Inspirada en [14].

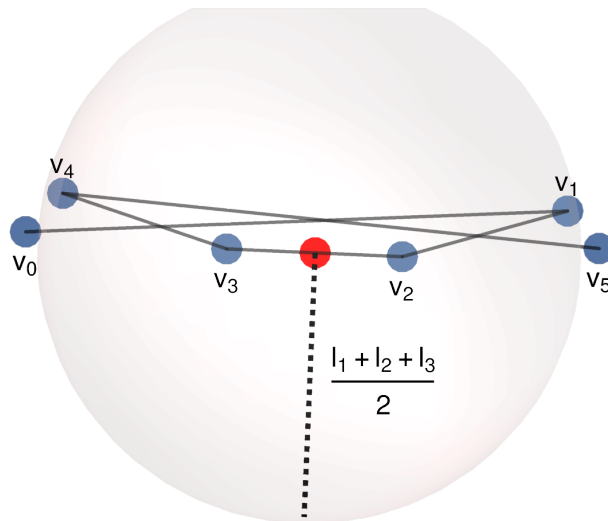


Figura 2.4: Movimiento que intenta llevar la cadena de la figura 2.3 a una configuración plana.

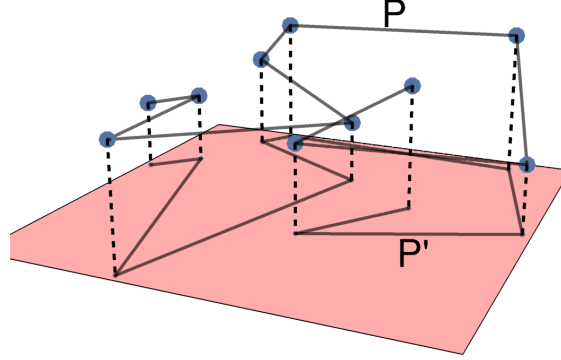


Figura 2.5: Cadena P con proyección sobre un plano P' sin intersección. Inspirada en [14].

$r > 0 \in \mathbb{R}$ tal que

$$r \leq \min_{\substack{0 \leq i \leq n \\ 1 \leq j \leq n \\ j \neq i, i-1}} \{d(\mathbf{v}_i, \mathbf{u}_j)\} \quad (2.5)$$

donde $d(\mathbf{v}, \mathbf{u})$ está definida como la distancia mínima entre el punto \mathbf{v} y el segmento \mathbf{u} . Coloquialmente, r es menor o igual que la distancia a cualquier vértice y un enlace que no sea adyacente a él. Es claro que si en cada vértice \mathbf{v}_i construimos un cilindro circular infinito C_i con eje paralelo al eje \mathbf{e}_z centrado en dicho vértice y con radio r , el cilindro no contendrá ningún otro vértice ni ningún segmento que no sea adyacente a \mathbf{v}_i . Más aún, los cilindros serán ajenos entre ellos.

Ahora, para llevar la cadena a la configuración canónica, primero reducimos el ángulo interno entre \mathbf{u}_1 y el eje de C_1 hasta que la distancia entre \mathbf{v}_0 y el eje mencionado sea igual a $\delta = r/n$. Notemos que dicho movimiento es posible ya que en la proyección 2D de la cadena solo estamos acortando el segmento \mathbf{u}'_i y, por la hipótesis de que la proyección de la cadena no tiene intersecciones, dicho movimiento no puede generar ninguna intersección con otro segmento de la cadena P' . Posterior a esto, rotamos \mathbf{u}_1 sobre el eje del cilindro para que quede contenida en el plano Π_2 que contiene al eje del cilindro y a \mathbf{u}_2 . Dicho desplazamiento se muestra en la figura 2.6. Ya realizado este proceso, lo seguimos repitiendo para \mathbf{u}_2 sobre el cilindro C_2 y así sucesivamente con todos los otros segmentos. Notemos que, como muestra la figura 2.7 en el i -ésimo paso tendremos i segmentos totalmente contenidos

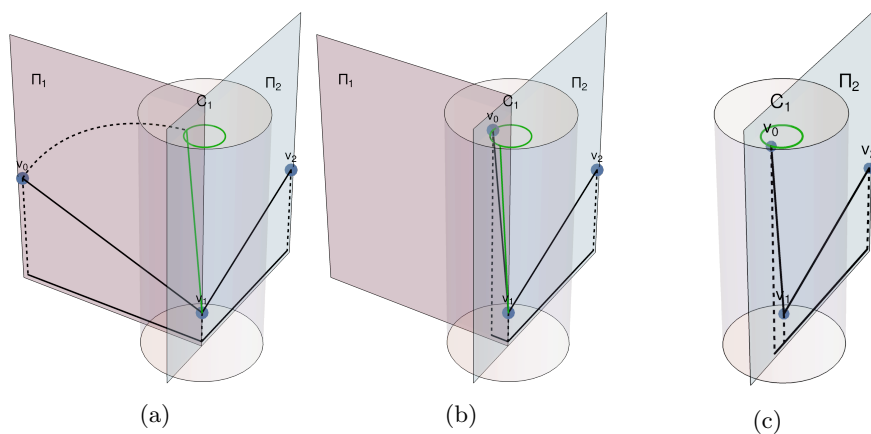


Figura 2.6: Paso inicial en la prueba del teorema 2.8. (a)→(b) primero rotamos v_0 hasta estar en el círculo verde de radio δ (b) → (c) rotamos v_0 sobre el eje del cilindro hasta estar en el plano Π_2 .

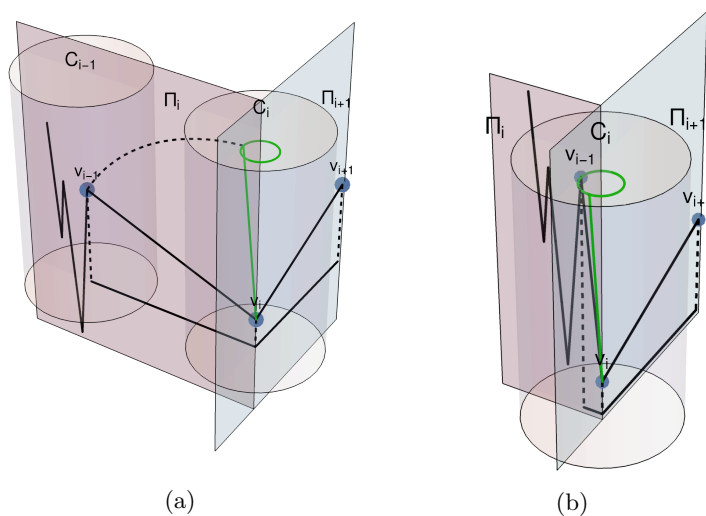


Figura 2.7: Paso intermedio en la prueba del teorema 2.8. Notemos como, al movernos de un cilindro a otro, la cadena siempre queda contenida dentro de dicho cilindro.

en el cilindro C_i y contenidos todos en un plano en forma de zigzag. Así, al realizar el procedimiento para el segmento C_n , ya tendremos toda la cadena en forma de zig zag contenida en un plano, sin intersecciones, por lo que podemos nuevamente aumentar los ángulos internos desde el mayor hasta el

menor para obtener la configuración canónica. ■

Corolario 2.9. *Sea P una cadena inscrita en la superficie de un polihedro convexo tal que no contiene intersecciones. Entonces la cadena no está encerrada.*

Demostración. Se puede realizar el mismo proceso del teorema 2.8 pero ahora los cilindros serán paralelos a la superficie de la cara del polihedro que contiene al vértice y aprovechar la hipótesis de que sea convexo para mover los segmentos evitando intersecciones con otras caras del polihedro. Si el polihedro se puede desdoblar para que esté contenido en un plano, entonces podemos primero desdoblar el polihedro y después es exactamente igual al original. ■

2.2.1. Cadenas de ángulo interno fijo

Un caso especial de particular interés por sus aplicaciones a biología son las cadenas en las que los ángulos internos θ_i están fijos pero los ángulos diedros ϕ_j pueden variar. Una cadena de longitud n de este estilo tiene tan solo $n - 2$ grados de libertad correspondientes a los ángulos diedros. Naturalmente, toda cadena encerrada con ángulos internos variables, como la de la figura 2.3, sigue estando encerrada cuando fijamos dichos ángulos. Así, resulta interesante encontrar una cadena que esté encerrada para ángulos internos fijos pero no cuando son variables.

La primera versión de una cadena así fue descubierta por Nadia Benbernou y su asesor Joseph O'Rourke, y se muestra en la figura 2.8. La primera mención de dicha cadena se encuentra en la tesis de licenciatura de Benbernou [24] y en el artículo que derivó de dicha tesis [25]. Sin embargo, en ambos trabajos no se exhibe la prueba de que la cadena esté encerrada y en su lugar se refiere a un manuscrito de O'Rourke que nunca fue publicado.

Teorema 2.10. *La cadena P de la figura 2.8 está encerrada para $l_1, l_6 \geq \sqrt{2}$.*

Una pregunta común que también se hace en torno a las cadenas de ángulo fijo es sobre si es posible modificar los ángulos diedros hasta llevar la cadena a una configuración plana, es decir, en la que todos los enlaces estén contenidos en el mismo plano.

Definición 2.5. Decimos que una cadena es **aplanable** (*flattenable*) sí y

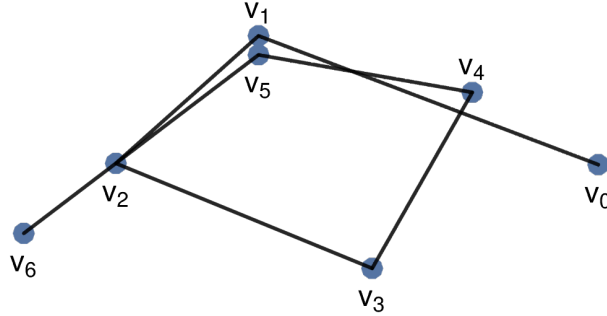


Figura 2.8: Cadena encerrada para ángulo interno fijo.

sólo si es posible reconfigurarla a la configuración en la que todos sus enlaces están contenidos en un plano.

La condición de que una cadena sea aplanable claramente es más débil que no ser encerrada. A diferencia del encerramiento, la posibilidad de aplanar una cadena no se puede conseguir con menos de 4 enlaces.

Teorema 2.11. *Cualquier cadena poligonal de ángulo interno fijo con longitud $n \leq 3$ es aplanable.*

Demostración. Es claro que el teorema se cumple para $n = 1, 2$. Para $n = 3$, sea $P = (\mathbf{v}_0, \dots, \mathbf{v}_3)$. Ya que \mathbf{u}_1 y \mathbf{u}_2 ya están contenidas en un plano, solo debemos añadir \mathbf{u}_3 a ese plano. Notemos que, ya que los ángulos internos están fijos, \mathbf{u}_3 solo cambiar de ángulo diedro, que se traduce en que \mathbf{u}_3 rote en un cono centrado en \mathbf{v}_2 con eje de simetría colineal a \mathbf{u}_2 .

Así, al rotar sobre dicho cono para integrarse al plano de \mathbf{u}_1 y \mathbf{u}_2 solamente podría colisionar con \mathbf{u}_1 . Sin embargo, ya que se puede rotar sobre el cono en dos direcciones, siempre es posible llevar a \mathbf{u}_3 a dicho plano. ■

Es posible exhibir una cadena poligonal de ángulo fijo con $n = 4$ que no es aplanable. Dicha cadena se muestra en la figura 2.9.

Teorema 2.12. *La cadena poligonal de la figura 2.9 no es aplanable cuando*

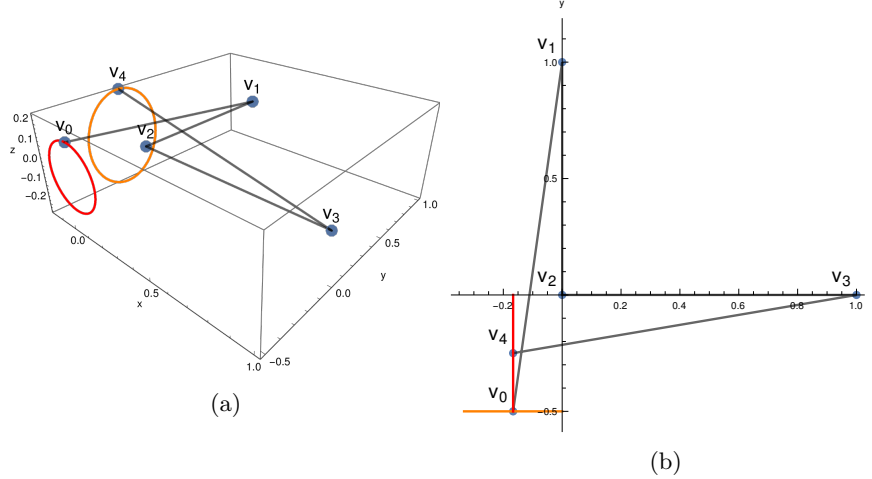


Figura 2.9: (a) Cadena poligonal no aplanable con $n = 4$. Explícitamente, los vértices son $\mathbf{v}_0 = (0, -1/2, \epsilon)$, $\mathbf{v}_1 = (0, 1, 0)$, $\mathbf{v}_2 = (0, 0, 0)$, $\mathbf{v}_3 = (1, 0, 0)$, $\mathbf{v}_4 = (-\epsilon, 0, 3\epsilon/2)$. Los círculos rojo y naranja muestran el posible movimiento de los vértices \mathbf{v}_0 y \mathbf{v}_4 (b) movimiento necesario para aplanar la cadena.

$\epsilon < 1/3$.

Demostración. Es claro que la única posible configuración de la cadena aplanada que no genera intersecciones es la que tiene al segmento \mathbf{u}_1 intersectado con el eje x en la parte negativa y al \mathbf{u}_4 con el eje y negativo. Sin pérdida de generalidad, supongamos que primero rotamos \mathbf{u}_1 para colocarlo en el plano XY .

Para rotar \mathbf{u}_4 ahí, ya que $\epsilon < 1/3$, entonces $3\epsilon/2 < 1/2$, por lo que \mathbf{u}_4 siempre intersectaría a \mathbf{u}_1 en el plano XY como muestra la figura 2.9(b). ■

No existe una respuesta sobre la complejidad de decidir si una cadena poligonal está encerrada. Sin embargo, decidir si una cadena es aplanable resulta ser un problema intratable en tiempo polinomial.

Lema 2.13. [26, 27] Sea $S = \{s_1, \dots, s_n\}$ un conjunto de enteros positivos. Entonces decidir si existen una partición $S_1, S_2 \subset S$ tal que $\sum_{a \in S_1} a = \sum_{b \in S_2} b$ es un problema NP duro.

Teorema 2.14. Decidir si una cadena poligonal con ángulos internos fijos

es aplanable es un problema NP duro.

Demostración. La prueba del teorema, siguiendo a [28], se hace mostrando que el problema de decisión es equivalente al problema del lema 2.13. Así, dado un conjunto S , construiremos una cadena poligonal de ángulo fijo P y probaremos que S se puede particionar como en el lema sí y sólo si la cadena P es aplanable.

Sin pérdida de generalidad, supongamos que $S = \{s_1, \dots, s_6\}$. Sea $\sigma = \sum_{s \in S} s$ y un $\epsilon > 0$, la primera parte que debemos construir de la cadena es la mostrada en la figura 2.10

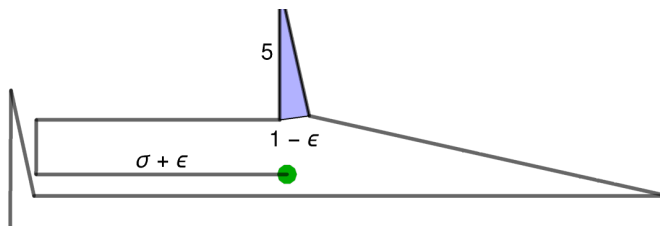


Figura 2.10: Primera parte de la cadena usada en el teorema 2.14.

Esta parte de la cadena la colocamos dentro del plano XY . Ahora, sobre el punto verde construimos la cadena mostrada en la figura 2.11 pero ahora sobre el plano XZ .

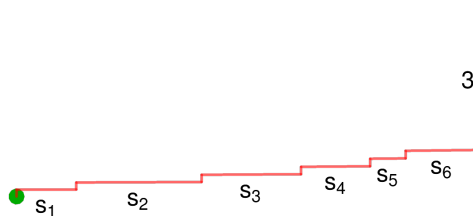


Figura 2.11: Segunda parte de la cadena usada en el teorema teorema 2.14. El último enlace tiene longitud 3.

El resultado final termina siendo algo parecido la figura 2.12

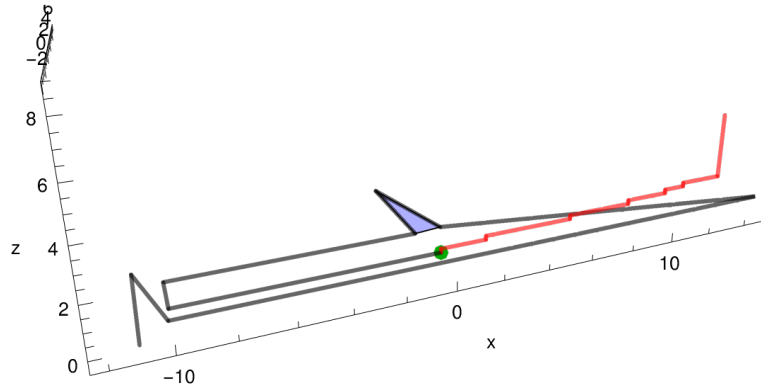


Figura 2.12: Cadena completa para el teorema 2.14.

Para poder aplanar la cadena, debido a la longitud y dirección del último enlace de la cadena roja, la única forma de hacerlo es colocando dicho enlace dentro del triángulo azul mostrado en la figura . Así, la parte negra de la cadena funciona como una cerradura que impide que la parte roja pueda pasar por otro lugar distinto al triángulo mencionado.

Por otro lado, la única forma en la que se puede colocar dicho enlace en el triángulo es encontrando una partición de S que permita colocar todos los enlaces rojos dentro del rectángulo de lado $\sigma + \epsilon$. Para el caso dibujado en las figuras, dicha partición es posible y este proceso se muestra en las figuras 2.13. Es claro que se puede generalizar para $|S| \geq 6$. Así, la figura puede aplanarse si y solo si existe una partición, por lo que los problemas son equivalentes, y, por el lema 2.13, NP duros. ■

2.3. Aplicación a la estructura proteínica

Una cuestión muy importante sobre las cadenas poligonales surge en su relación con las proteínas. Una proteína es una macromolécula biológica que se puede describir de forma simple como una sucesión finita de aminoácidos unidos entre sí por un enlace peptídico. Un aminoácido es una molécula química que, como muestra la figura 2.14 consta de un carbón central, también llamado **carbón alfa** y denotado C_α , unido a un grupo carboxilo (CO_2^-), un grupo amino NH_3^+ , un hidrógeno H y una cadena lateral denotada con una letra R . Los aminoácidos se pueden unir entre sí mediante un tipo de enlace covalente conocido como enlace péptido. El enlace se realiza

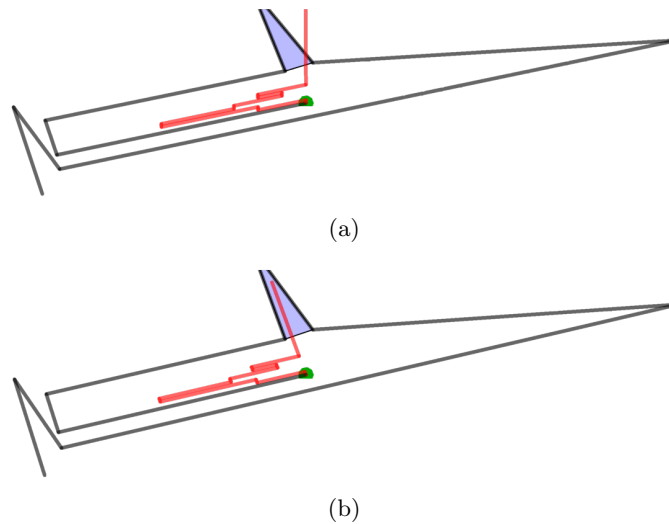


Figura 2.13: Movimiento para aplanar la cadena de la figura 2.11 (a) se colocan los enlaces de la partición en el rectángulo del cerrojo. (b) Se rota para colocar el enlace rojo en el triángulo azul.

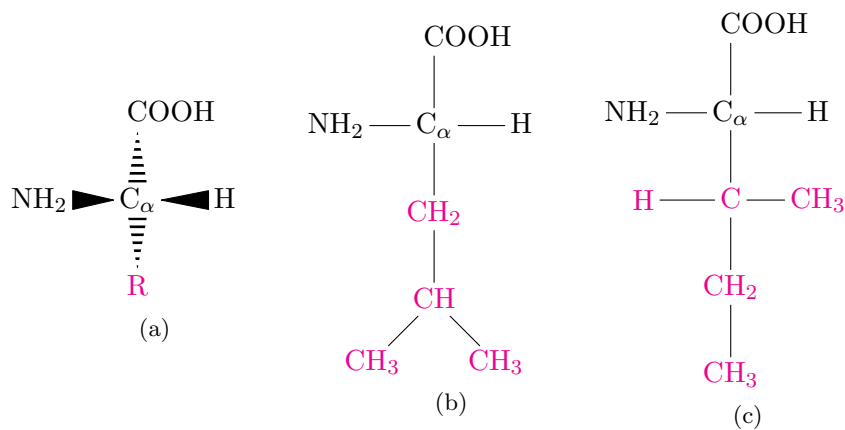


Figura 2.14: (a) Diagrama esquemático de un aminoácido. (b,c) Dos aminoácidos, leucina e isoleucina, con sus respectivas cadenas laterales.

entre el carbono del grupo carboxilo y el nitrógeno del grupo amino. La figura 2.15 muestra la reacción química y la topología de dicho enlace.

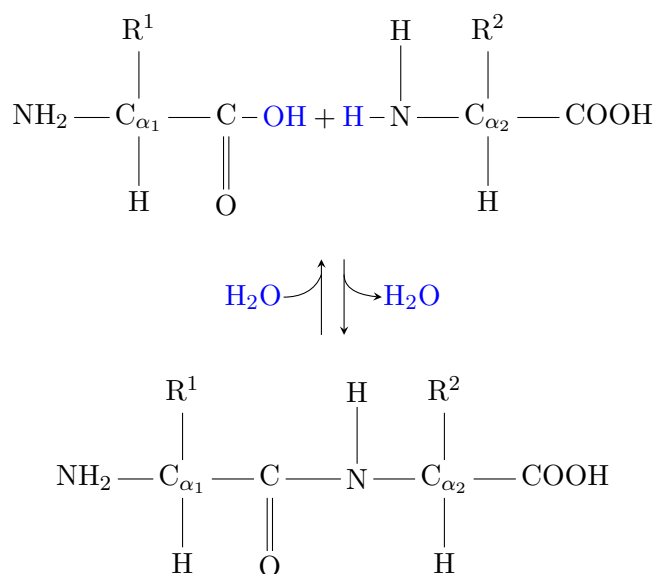


Figura 2.15: Reacción de condensación en la que se genera un enlace péptido entre dos aminoácidos.

Así, es posible formar cadenas (también llamadas polímeros) de aminoácidos unidos mediante enlaces péptidos. A dichas cadenas se les conoce como **proteínas**. Biológicamente, las proteínas son moléculas que realizan funciones sumamente importantes en todos los organismos vivos, por lo que su estudio resulta particularmente importante.

Es sabido que el correcto funcionamiento de una proteína depende de que pueda plegarse desde una estructura 3D arbitraria hasta su **estructura nativa**, una estructura particular en 3D que resulta ser la más estable, termodinámicamente hablando, para dicha proteína. Más aún, es sabido que dicha estructura nativa está únicamente determinada por los aminoácidos que componen la proteína y por otros factores fisicoquímicos como la temperatura y el pH del medio en el que se realice el plegamiento [29].

Así, existe un gran interés en estudiar el proceso mediante el cual las proteínas alcanzan su estructura nativa a partir de una arbitraria. La gran mayoría de modelos físicos para dicho procesos son sumamente complejos, pues el número de configuraciones que puede tener una proteína generalmente crece exponencialmente [30]. Sin embargo, el tiempo que le toma a una proteína llegar a su estructura nativa varía entre $10^{-6} - 10^2$ segundos. A encontrar un modelo adecuado que pueda explicar en términos físicos cómo

se realiza el plegamiento de una proteína se le conoce como el **problema del plegamiento de proteínas**.

Independientemente de los sofisticados métodos de simulación física que permiten atacar el problema del plegamiento de proteínas, existe una idea sencilla que podría dar una explicación sensata a dicho problema. Antes de proceder a eso, debemos dar algunas definiciones. Generalmente, el enlace peptídico y los otros enlaces covalentes entre C_α y el carbono del grupo CO_2^- y el nitrógeno de NH_3^+ tienen una longitud constante, al igual que los ángulos entre los enlaces, cuyos valores se muestran en la figura 2.16.

Es posible probar que el enlace péptido es plano, es decir, que todos los átomos $C_{\alpha i} - C_i - N_i - C_{\alpha i+1}$ están todos en un mismo plano para todo i . Así, tiene sentido definir la **cadena principal** de una proteína como la cadena poligonal que une entre sí a los C_α . En la figura 2.16 se muestra dicha cadena.

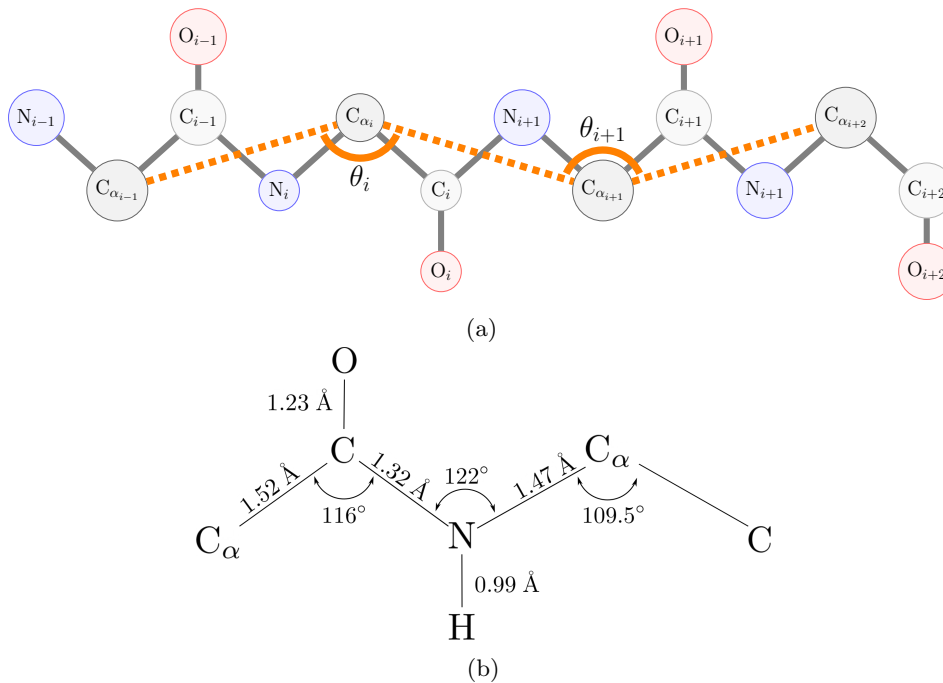


Figura 2.16: (a) Esquema de una proteína. La cadena principal se muestra punteada en color naranja. (b) Distancias características de los enlaces en una proteína [30, 31, 14].

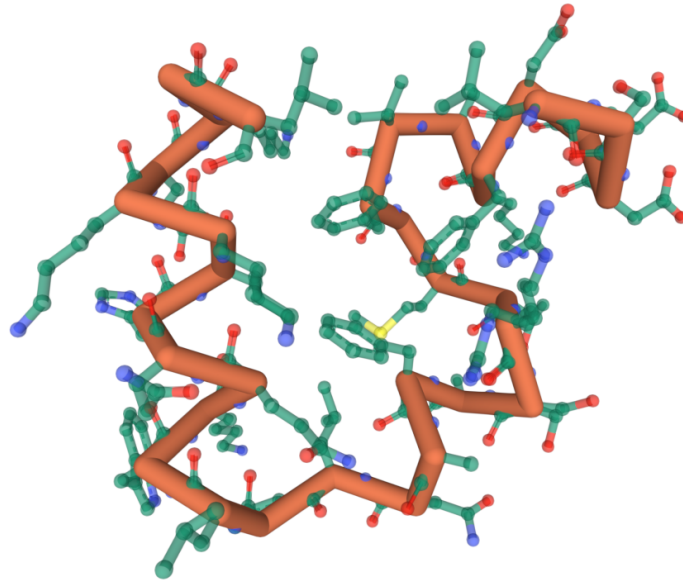


Figura 2.17: Estructura 3D de la proteína 3TJW [32]. La cadena principal se muestra en color naranja.

Es claro que, debido a las propiedades discutidas sobre los enlaces covalentes, la cadena principal será una cadena poligonal con enlaces de longitud fija y ángulos internos fijos y obtusos. En general, la propiedad de tener ángulos internos no agudos es suficientemente importante para tener su propia definición.

Definición 2.6. Dado $0 \leq \alpha \leq \pi/2$, decimos que una cadena poligonal de ángulo fijo es una $(\leq)\alpha$ -cadena sí y sólo si todos sus ángulos internos satisfacen que $0 \leq \pi - \theta_i \leq \alpha$.

En un plano más realista, las proteínas se crean en una parte de la célula llamada ribosoma. Aunque la geometría del ribosoma en general puede ser muy exótica, un modelo simple [33] lo plantea como un cono con un ángulo de apertura α . Es posible realizar un esquema básico de como es que un ribosoma cónico generaría cadenas poligonales representativas de la cadena poligonal de una proteína. La figura 2.18 muestra un ribosoma real y el modelo planteado. Las cadenas poligonales producidas por un ribosoma cónico de ángulo α resultan ser $(\leq)\alpha$ -cadenas. En particular, tienen una

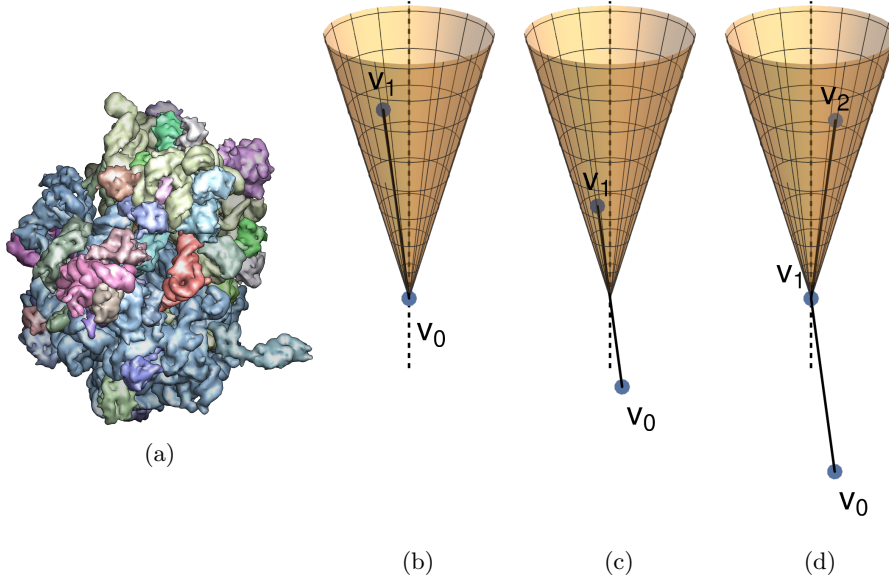


Figura 2.18: (a) Estructura 3D de un ribosoma, tomada de [34] (b,c,d) Modelo simple de un ribosoma cónico produciendo una cadena poligonal [33].

propiedad más importante. Antes que nada, haremos la distinción entre una cadena y su espacio de configuraciones. Recordemos que, usando coordenadas internas, podemos definir P una $(\leq)\alpha$ cadena poligonal como un punto $P = (l_1, \dots, l_n, \theta_1, \dots, \theta_{n-1}) \in [0, \infty)^n \times [\pi/2, \pi]^{n-1}$. El espacio de configuraciones de dicha cadena son todos los posibles valores de sus ángulos diedros $\Omega = [0, 2\pi]^{n-2}$. Así, denotamos una configuración $\omega = (\phi_1, \dots, \phi_{n-2}) \in \Omega$.

Teorema 2.15. [35] *Una configuración de una $(\leq)\alpha$ -cadena es producible por un ribosoma cónico sí y sólo si dicha configuración es aplanable.*

El teorema 2.15 es sumamente importante ya que identifica que la clase de $(\leq)\alpha$ -cadenas que se pueden producir en el modelo ribosomal con la clase de $(\leq)\alpha$ -cadenas aplanables. La prueba del teorema escapa al alcance de este trabajo ya que no nos interesa examinar en detalle dicho modelo de producción. Sin embargo, hay algunos otros resultados relevantes relacionados con la probabilidad de que una cadena esté encerrada dada. Presentamos algunas definiciones relacionadas a Probabilidad, refiriendo al lector a [36] para una revisión de los conceptos básicos.

Definición 2.7. Sea \mathbf{X} un vector aleatorio con distribución asociada de

forma única al espacio de probabilidad $(\mathbb{R}^d, \mathbb{B}(\mathbb{R}^d), \mathbb{P})$ y sea μ la medida de Lebesgue sobre $\mathbb{B}(\mathbb{R}^d)$. Decimos que la distribución asociada a \mathbf{X} es **regular** sí y sólo sí existe un conjunto abierto, llamado dominio, $D \subset \mathbb{R}^d$ tal que para todo abierto $A \subset D$ si $\mu(A) > 0$ entonces $\mathbb{P}(A) > 0$ y para $B \in \mathbb{R}^d \setminus D$ tenemos $\mathbb{P}(B) = 0$.

Definición 2.8. Sean \mathcal{L} , Θ y Φ distribuciones de probabilidad regulares con soporte $[0, \infty)$, $[\pi/2, \pi]$ y $[0, 2\pi)$, respectivamente. Denotamos con $\mathcal{P}_n^{\Theta, \mathcal{L}}$ a la distribución de probabilidad de una $(\leq)\alpha$ -cadena poligonal de longitud n tal que $l_i \sim \mathcal{L}$ y $\theta_i \sim \Theta$ y son independientes entre sí.

Definición 2.9. Dada una $(\leq)\alpha$ -cadena P de longitud n , si generamos $n-2$ ángulos diedros independientes de la distribución Φ entonces obtenemos la que la distribución todas sus configuraciones $\mathcal{Q}^{P, \Phi}$. Si quitamos del dominio de $\mathcal{Q}^{P, \Phi}$ las configuraciones que presentan intersecciones entre enlaces y reescalamos la medida de probabilidad, entonces denotamos a la distribución resultante como $\mathcal{Q}^{P, \Phi}$.

Dadas estas definiciones, podemos enunciar los siguientes resultados, todos tomados de [35]

Lema 2.16. *El subespacio de las configuraciones sin intersección de una cadena P es abierto.*

Demostración. Notemos que la condición de que una cadena de longitud n no tenga intersecciones se puede expresar en $l \in \mathcal{O}(n^2)$ restricciones de la forma

$$g_k(\phi_1, \dots, \phi_{n-2}) > 0, \quad (2.6)$$

donde g_k es un polinomio en $\sin(\phi_i)$ y $\cos(\phi_i)$. La restricción de cada g_k define un conjunto abierto en el espacio de configuraciones, por lo que la intersección finita de todas las restricciones sigue siendo un conjunto abierto. ■

Definición 2.10. Decimos que una configuración Q de una cadena P está encerrada sí y sólo sí no puede llevarse a una configuración plana.

Lema 2.17. *Dada una distribución regular de configuraciones $\mathcal{Q}_n^{P, \Phi}$ sobre una cadena encerrada P , si hay una configuración encerrada en el dominio de \mathcal{Q} , entonces la probabilidad de que una configuración aleatoria Q de P este encerrada es mayor o igual a una constante $c > 0$.*

Demostración. Sea $Q' \in \mathcal{Q}$ la configuración encerrada mencionada y D el dominio de \mathcal{Q} . Sea ahora C la componente del espacio de configuraciones de P que contiene a Q' y $G = C \cap D$. Por el lema 2.16, sabemos que C es abierto y D es abierto por definición, por lo que G es abierto también. Así, sabemos que existe un ϵ tal que la bola $B_\epsilon(Q) \in \mathbb{R}^{n-2}$ cumple que para todo $Q'' \in B_\epsilon(Q)$, Q'' está encerrada. Por ser \mathcal{Q} regular, es claro que $\mathbb{P}[Q \in B_\epsilon(Q)] = c > 0$. Así, por monotonicidad de la probabilidad, tenemos que $\mathbb{P}[Q \text{ este encerrada}] \geq c > 0$. ■

Lema 2.18. *Dada una distribución regular de configuraciones $\mathcal{P}_n^{\mathcal{L}, \Theta}$, si hay una cadena encerrada en el dominio de \mathcal{P} entonces la probabilidad de que una cadena aleatoria P este encerrada es mayor o igual a una constante $d > 0$*

Demostración. Primero notamos que el espacio de todas las cadenas y sus configuraciones se puede denotar como $C = [0, \infty)^n \times [\pi/2, \pi]^{n-1} \times [0, 2\pi)^{n-2}$. Como en el lema 2.16, notemos que la restricción de que una cadena esté encerrada se puede expresar con $\mathcal{O}(3n - 3)$ restricciones de la forma

$$h_l(l_1, \dots, l_n, \theta_1, \dots, \theta_{n-1}, \phi_1, \dots, \phi_{n-2}) > 0, \quad (2.7)$$

con h_l un polinomio en $l_i, \cos(\theta_j), \sin(\theta_j), \cos(\phi_k), \sin(\phi_k)$. Cada h_l define un conjunto abierto en el espacio de configuraciones. La intersección de todas estas restricciones resulta en un conjunto abierto $L \subset C$. Sea ahora L' la proyección de L sobre $[0, \infty)^n \times [\pi/2, \pi]^{n-1}$. Notemos que L' sigue siendo abierto ya que las proyecciones son funciones abiertas [37].

De forma análoga a la prueba del lema 2.17, dada una cadena poligonal encerrada P' existe un $\epsilon > 0$ tal que $B_\epsilon(P')$ está contenida en la intersección de la componente de L' que contiene a P' y el dominio de \mathcal{P} tal que toda $P'' \in B_\epsilon(P')$ está encerrada. Ya que la distribución \mathcal{P} es regular, tenemos que $\mathbb{P}[P \text{ está encerrada}] \geq \mathbb{P}[P \in B_\epsilon(P')] = d > 0$. ■

Todos estos lemas nos permiten probar el siguiente teorema:

Teorema 2.19. *Sea P_n una cadena aleatoria tomada de la distribución $\mathcal{P}_n^{\mathcal{L}, \Theta}$. Si existe una cadena encerrada en el dominio de $\mathcal{P}_n^{\mathcal{L}, \Theta}$ para alguna n , entonces*

$$\lim_{n \rightarrow \infty} \mathbb{P}[P_n \text{ no está encerrada}] = 0. \quad (2.8)$$

Más aún, si Q_n es una configuración aleatoria tomada de $\mathcal{Q}^{P_n, \Phi}$ entonces

$$\lim_{n \rightarrow \infty} \mathbb{P}[Q_n \text{ es aplanable}] = 0. \quad (2.9)$$

Ambos límites convergen de forma geométrica

Demostración. Sea $m \in \mathbb{N}$ tal que existe una cadena encerrada de longitud m . Sea ahora un $n \geq m \in \mathbb{N}$ y una cadena $P_n \sim \mathcal{P}_n^{\mathcal{L}, \Theta}$. Dividamos la cadena en $\lfloor n/m \rfloor$ subcadenas de longitud m . Claramente tenemos que cada subcadena $P_m^{(i)} \sim \mathcal{P}_m^{\mathcal{L}, \Theta}$. Notemos que la cadena P_n no está encerrada sí y sólo si cada una de sus $\lfloor n/m \rfloor$ subcadenas no está encerrada. Por el lema 2.17, sabemos que $\mathbb{P}[P_m^{(i)} \text{ está encerrada}] \geq c$. Así, por independencia, sabemos que

$$\begin{aligned} \mathbb{P}[P_n \text{ no está encerrada}] &= \prod_{i=1}^{\lfloor n/m \rfloor} \mathbb{P}[P_m^{(i)} \text{ no está encerrada}] \\ &= \prod_{i=1}^{\lfloor n/m \rfloor} (1 - \mathbb{P}[P_m^{(i)} \text{ está encerrada}]) \\ &\leq \prod_{i=1}^{\lfloor n/m \rfloor} (1 - c) \\ &= (1 - c)^{\lfloor n/m \rfloor}. \end{aligned} \quad (2.10)$$

Tomando $n \rightarrow \infty$ de ambos lados de la desigualdad, encontramos que $\mathbb{P}[P_n \text{ no está encerrada}] \rightarrow 0$. De forma análoga, notemos ahora que, por independencia entre los ángulos diedros y los demás parámetros, y por los lemas 2.17, 2.18, tenemos:

$$\begin{aligned} \mathbb{P}[Q_m \sim \mathcal{Q}_m^{P_m, \phi} \text{ esté encerrada}] &= \mathbb{P}[P_m \text{ esté encerrada}] \mathbb{P}[Q_m \text{ esté encerrada}] \\ &\geq cd > 0. \end{aligned} \quad (2.11)$$

De forma análoga, al dividir Q_m en $\lfloor n/m \rfloor$ subcadenas, se tiene que

$$\begin{aligned}
\mathbb{P}[Q_n \text{ aplanable}] &< \mathbb{P}[Q_n \text{ no esté encerrada}] \\
&= \prod_{i=1}^{\lfloor n/m \rfloor} \mathbb{P}[Q_m^{(i)} \sim \mathcal{Q}_m^{P,\phi} \text{ no está encerrada}] \\
&= \prod_{i=1}^{\lfloor n/m \rfloor} (1 - \mathbb{P}[Q_m^{(i)} \sim \mathcal{Q}_m^{P,\phi} \text{ está encerrada}]) \quad (2.12) \\
&\leq \prod_{i=1}^{\lfloor n/m \rfloor} (1 - cd) \\
&= (1 - cd)^{\lfloor n/m \rfloor}.
\end{aligned}$$

Nuevamente, tomando $n \rightarrow \infty$ de ambos lados encontramos que $\mathbb{P}[Q_n \text{ sea aplanable}] \rightarrow 0$ ■

Estos resultados son de suma importancia pues nos permiten establecer que, si existe una $(\leq)\alpha$ cadena encerrada de cierta longitud, la probabilidad de que otra cadena aleatoria de mayor longitud esté encerrada se acerca a 1 de forma geométrica. Así, probar que existen cadenas encerradas implica, de forma indirecta, que es muy probable que las cadenas principales de las proteínas también lo estén.

Esto tiene una implicación directa en los modelos utilizados en física estadística para modelar las proteínas, pues implica que el espacio configuracional de las proteínas no es tan grande como se ha estimado y abre la posibilidad a realizar algunas simplificaciones o reducciones en los métodos computacionales que mejoren su eficiencia.

2.4. Teoría de nudos y funcionales de energía

Existe una relación entre el problema de decidir si una cadena poligonal está encerrada o no con la teoría de nudos, una rama de la topología que estudia las figuras que coloquialmente se pueden entender como nudos sobre un círculo. Podemos dar una definición formal de algunos conceptos. Remitimos al lector ajeno a la topología básica a [38].

Definición 2.11. Un **nudo** (*knot*) es un encaje, i.e. una función continua e inyectiva, $f : S^1 \rightarrow \mathbb{R}^3$ de clase C^1 . Suponemos que su derivada f' nunca se

anula. Los nudos como curvas cerradas están orientadas. Cuando hablamos de un nudo en abstracto, por lo general nos referimos a su imagen en $K = f(S^1)$

Definición 2.12. Dos nudos f y g son **equivalentes** o isotópicos si existe un homeomorfismo $h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ que preserve orientación tal que $g = h \circ f$. La equivalencia de nudos es una relación de equivalencia que define clases de equivalencia, denotadas $[K]$.

Algunos textos [39, 40] también piden que la función f sea una función lineal a pedazos para evitar tratar con nudos con cruces infinitos. Más aún, las definiciones de equivalencia pueden ser distintas también. La figura 2.19 muestra algunos ejemplos de nudos.

El problema fundamental que estudia la teoría de nudos es el **problema de la equivalencia de nudos**: dados dos nudos K, L , ¿son equivalentes?. Un caso particular de dicho problema es cuando $L = S^1$, es decir, cuando L es un simple círculo o un nudo trivial. La respuesta a ambos problemas, de forma similar al problema de encerramiento, se puede dar de distintas formas: podemos responder si es posible decidir el problema, encontrar su complejidad computacional como función de algún parámetro y, en el mejor de los casos, dar un algoritmo para él. Se sabe que el problema de reconocer si un nudo es trivial es al menos de complejidad NP [41], por lo que el problema de equivalencia de nudos debe tener una complejidad mayor a esa. La forma tradicional de atacar el problema de equivalencia, como es común en topología, es mediante el cálculo de invariantes topológicos. Entre los invariantes más utilizados caben destacar el polinomio de Jones, el polinomio de Alexander y el polinomio HOMFLY [20]. Una de las grandes ventajas de estos polinomios es que, a pesar de ser invariantes, pueden calcularse a partir de un dibujo del nudo [20].

Sin embargo, independientemente del uso de invariantes, una pregunta que puede surgir sobre la clase de equivalencia de los nudos es: ¿cómo sería posible escoger un nudo representante de una clase de equivalencia? De existir, ¿cómo podríamos transformar continuamente un nudo arbitrario a dicha representación canónica? La posibilidad de llevar un nudo a una representación canónica nos daría una forma de responder al problema de equivalencia pues bastaría ver si dos nudos llegan a la misma configuración canónica.

Una forma de encontrar una configuración canónica de un nudo consiste en, dado un funcional sobre la clase de equivalencia de un nudo $E : [K] \rightarrow \mathbb{R}$

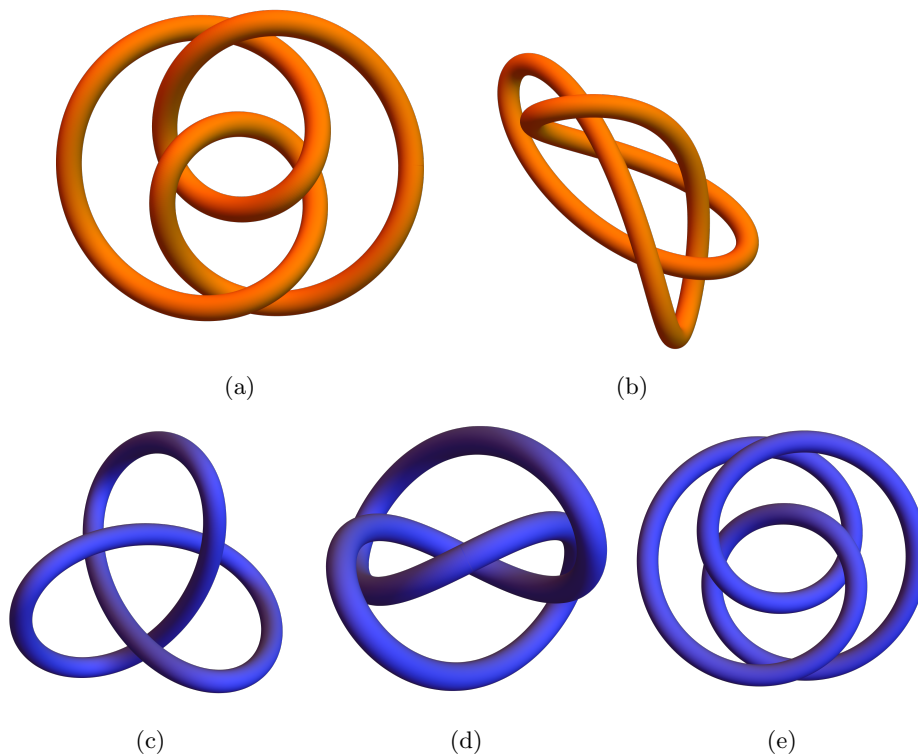


Figura 2.19: Distintas representaciones de algunos nudos. (a,b) pertenecen a la misma clase de equivalencia mientras que (c,d,f) pertenecen a otra.

que cumpla ciertas condiciones, definir la configuración canónica como el mínimo global de dicho funcional. Esta propuesta fue realizada por O'Hara [42, 16]. Al funcional E se le conoce como **funcional de energía**. Podemos definirlo junto con sus condiciones y propiedades.

Definición 2.13. Un **funcional de energía** es una función $E : C^1(S^1, \mathbb{R}^3) \rightarrow [0, \infty]$ que cumple que para toda sucesión $K_{n \in \mathbb{N}} \subset C^1(S^1, \mathbb{R}^3)$ que converge de forma uniforme a una curva no inyectiva $K_\infty \in C^1(S^1, \mathbb{R}^3)$ se cumple que $E(K_n) \rightarrow \infty$ para $n \rightarrow \infty$.

Otras propiedades importantes que generalmente se le piden a las energías funcionales es que tiendan a ∞ cuando un nudo empieza a contraerse en un punto. Los primeros ejemplos de funcionales de energía E_1^α se inspiraron en el potencial electrostático de un nudo si fuera un objeto cargado con densidad

lineal de carga. Sin pérdida de generalidad, supongamos que $\|f'(u)\| = 1$. Así, definimos de forma intuitiva

$$E_1^\alpha(f) = \int_{S^1} \int_{S^1} \frac{1}{\|f(u) - f(v)\|^\alpha} du dv, \quad (2.13)$$

Esta expresión diverge debido a que la interacción electrostática diverge cuando $u \rightarrow v$. Si embargo, es posible regularizar dicha expresión con otro término para evitar dicha divergencia

Definición 2.14. La **energía de Möbius** $E_1^{\alpha,p}$ está definida como

$$E_1^{\alpha,p}(f) = \int_{S^1} \int_{S^1} \left(\frac{1}{\|f(u) - f(v)\|^\alpha} - \frac{1}{D_f(u,v)^\alpha} \right) du dv, \quad (2.14)$$

donde $\alpha, p \in (0, \infty)$ y $D_f(u, v)$ es la distancia sobre la curva f de los puntos $f(u)$ y $f(v)$. Por definición, siempre se cumple que $\|f(u) - f(v)\| \leq D_f(u, v)$ por lo que $E_1^{\alpha,p}(f) \in [0, \infty]$.

Teorema 2.20. [17] *La energía de Möbius es invariante bajo transformaciones de Möbius. Más aún, para $\alpha p \geq 2$, la función es un funcional de energía.*

Una desventaja de la energía de Möbius es que no hay una divergencia a la hora de apretar un nudo hasta cerrarlo por lo que no son las más aptas desde un punto de vista de optimización numérica [18].

Un conjunto de funcionales que si divergen cuando se aprieta un nudo son las energías de punto tangente

Definición 2.15. La **energía de punto tangente** TP_b^a está definida como

$$TP_b^a(f) = \int_{S^1} \int_{S^1} \frac{\|P_{f'(u)}^\perp(f(u) - f(v))\|^a}{\|f(u) - f(v)\|^b} du dv, \quad (2.15)$$

con

$$P_{f'(u)}^\perp(\mathbf{x}) = \mathbf{x} - (\mathbf{x} \cdot f'(u))f'(u) \quad (2.16)$$

la proyección de \mathbf{x} sobre el plano con normal $f'(u)$ y $b, a \geq 0$.

Para el caso donde $b = 2a$, la expresión

$$\|P_{f'(u)}^\perp(f(u) - f(v))\|^a / \|f(u) - f(v)\|^{2a}$$

es la a -ésima potencia del recíproco del diametro del círculo tangente a $f(u)$ que pasa por $f(v)$. Una expresión más sencilla se puede obtener recordando que una forma de expresar a $P_{f'(u)}^\perp(f(u) - f(v))$ es como

$$P_{f'(u)}^\perp(f(u) - f(v)) = T(u) \times (f(u) - f(v)), \quad (2.17)$$

por lo que TP_b^a se expresa de la siguiente manera:

$$TP_b^a(f) = \int_{S^1} \int_{S^1} \frac{\|T(u) \times (f(u) - f(v))\|^a}{\|f(u) - f(v)\|^b} du dv. \quad (2.18)$$

Teorema 2.21. [17] *La energía de punto tangente TP_b^a es un funcional de energía para $b, a \geq 1$ con $b \in (a + 2, 2a + 1)$. Más aún, la energía tiene la propiedad de diverger cuando se aprieta un nudo.*

La relación que se puede establecer entre la teoría de nudos y las cadenas poligonales es la siguiente. Para encontrar la configuración representativa de un nudo mediante minimización de energía, generalmente debemos de plantear el problema de forma numérica: el nudo se discretiza como una curva lineal a pedazos con n nodos, convirtiéndose en un vector $x \in \mathbb{R}^{3n}$. Así, el funcional de energía se convierte en una función $E : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ y podemos aplicar alguno de los muchos algoritmos de optimización numérica para encontrar un mínimo del funcional.

Realizar esta optimización no es un asunto trivial pues, independientemente de que tengamos una energía de nudo que diverja en las intersecciones y al apretar un nudo, el algoritmo de optimización debe ser capaz de evitar los movimientos que generarían una intersección en la cadena y de encontrar eficazmente un mínimo global.

2.5. Encierro de cadenas mediante minimización de energía

A la hora de discretizar un nudo en una curva lineal a pedazos, lo convertimos en una cadena poligonal cerrada, es decir, con el último vértice igual al primero. Así, si fijamos un valor de n y discretizamos todos los nudos para dicho valor, las clases de equivalencia de los nudos representarán distintas componentes conexas del espacio de configuraciones de dicha cadena cerrada. Así, determinar la clase de equivalencia de un nudo en su versión

discreta es equivalente a revisar el espacio de configuraciones de una cadena poligonal cerrada. Utilizando este paralelismo, en principio, **podríamos utilizar métodos de minimización de energía para explorar si es posible llevar una cadena poligonal a una configuración dada.**

Notemos que una diferencia fundamental es que las cadenas poligonales ya son un objeto compuesto de partes discretas. Por el contrario, un nudo topológico en principio es un objeto continuo que podemos discretizar en una cantidad arbitraria de valores. Esto presenta una dificultad, pues en la literatura [20, 18] se observa que la eficacia de los algoritmos de minimización en teoría de nudos tiene una dependencia grande en la cantidad de puntos utilizados en la discretización de un nudo. En particular, es posible observar que los mejores resultados se obtienen para $n > 100$. Esto provoca un problema a la hora de querer simular una cadena poligonal de tamaño pequeño como las exhibidas en la sección 2.2.

Independientemente de esto, el objetivo de este trabajo es explorar la posibilidad de llevar cadenas poligonales a una configuración canónica mediante un proceso de minimización de energía inspirado en la teoría de nudos y ver si los resultados obtenidos por medio de simulación numérica son consistentes con los resultados analíticos existentes. En el siguiente capítulo abordaremos las herramientas necesarias para realizar dicha simulación numérica.

Capítulo 3

Simulación numérica de cadenas poligonales

Inspirados en las metodologías de teoría de nudos para intentar reducir nudos a una configuración sencilla para clasificarlos, atacaremos el problema de cuándo es que una cadena poligonal está en una configuración encerrada al intentar llevarla desde una configuración en la que presente un nudo o alguna otra dificultad a una configuración canónica escogida por nosotros.

Para esto, utilizaremos distintas funciones de energía definidas sobre dicha cadena poligonal e intentaremos llevar la cadena poligonal a una configuración que corresponda al mínimo global de la energía. La sucesión de cadenas que se aproximan al mínimo global deberán cumplir con las restricciones que deseemos: que las longitudes de sus enlaces sean constantes y, en algunos casos, que sus ángulos internos también lo sean. Así, solo hay dos cosas que podemos modificar de una cadena poligonal: un ángulo diedro o interno. Esto implica que nuestro proceso de minimización requiere que, cuando busquemos generar nuevas configuraciones a partir de una existente al aumentar o reducir un ángulo de la cadena, debemos verificar que dicha modificación se puede hacer sin generar una intersección entre los enlaces de la cadena.

3.1. Descripción general

Para poder realizar el procedimiento que lleva a una cadena poligonal de una configuración a otra solamente modificando ángulos, es necesario verificar cuando se puede modificar uno de estos ángulos sin generar una intersección en la cadena.

Notemos que dicha modificación solo depende de la posible intersección en la cadena ya que en principio nuestros grados de libertad son los ángulos de la cadena y, al ser independientes entre sí, pueden modificarse sin generar configuraciones no válidas. La única restricción que tiene la cadena es la longitud de los enlaces y las intersección. Ya que cambiar un ángulo no cambia longitudes, solo debemos checar posibles intersecciones.

3.2. Intersección entre un segmento y la superficie generada por un segmento rotante

Ya que un ángulo diedro esta definido por los dos planos generados AB , AC por los segmentos de recta contiguos \mathbf{u}_{k-1} , \mathbf{u}_k , \mathbf{u}_{k+1} definidos por los cuatro vértices $\mathbf{v}_{k-2}\mathbf{v}_{k-1}\mathbf{v}_k\mathbf{v}_{k+1}$ de una cadena, es claro que, al ser el segmento \mathbf{u}_k común a ambos planos, el aumento del ángulo diedro ABC por un valor ϕ es equivalente a la rotación de todos los vértices \mathbf{v}_l con $l > k$ alrededor de la recta que contiene al segmento \mathbf{u}_k por un ángulo ϕ .

Así, las intersecciones que pueden existir al cambiar un ángulo diedro son las intersecciones entre segmentos de recta rotados y segmentos de recta estáticos. Por simplicidad, notemos que podemos restar \mathbf{v}_k a todos los vértices y rotar el sistema el mismo ángulo que el ángulo interno entre \mathbf{e}_z y \mathbf{u}_k en la dirección del vector $\mathbf{e}_z \times B$ para que el segmento \mathbf{u}_k quede paralelo a \mathbf{e}_z y sea más sencilla la descripción matemática de la rotación de los vértices.

Análogamente, cambiar el k -ésimo ángulo interno de la cadena, que se da entre los vértices $\mathbf{v}_{k-1}\mathbf{v}_k\mathbf{v}_{k+1}$, es equivalente a rotar todos los vértices \mathbf{v}_l con $l > k$ alrededor de la recta que contiene al vector $\mathbf{n}_k = \mathbf{u}_k \times \mathbf{u}_{k+1}$. Nuevamente, por simplicidad podemos rotar la cadena para que dicho vector esté en la misma dirección que el vector \mathbf{e}_z .

Para todo lo que continúa, supondremos que estamos en el marco de referencia en el cual el cambio de ángulo diedro es equivalente a la rotación de algunos vértices alrededor del eje z . Decidir

si es posible realizar el cambio de ángulo es equivalente a decidir si la superficie generada por los segmentos que giran tiene intersección con los segmentos que se quedan estáticos. Para decidir sobre la posibilidad de esta intersección, primero debemos describir matemáticamente el tipo de superficie que genera un segmento de recta al ser rotado un ángulo α sobre el eje z .

Dicha superficie cambiará drásticamente dependiendo si alguno de los segmentos está contenido en un plano paralelo al XY , por lo que es pertinente tomar casos.

3.2.1. Caso 1: ningún segmento está contenido en un plano paralelo al XY .

Podemos expresar analíticamente de forma sencilla la superficie generada, lo cual también nos será útil para verificar las intersecciones:

Lema 3.1. *La superficie de revolución generada por girar un segmento de recta no contenido en un plano paralelo al plano XY sobre el eje z satisface la ecuación*

$$\rho^2(z) = cz^2 + 2rz + d, \quad (3.1)$$

para algunos $c, r, d \in \mathbb{R}$ con $\rho^2(z) = x^2 + y^2$.

Demostración. Sean q_1, q_2 los extremos de la recta, notemos que podemos parametrizar el segmento de recta como:

$$\begin{aligned} x_q(s) &= q_{1x} + s(q_{2x} - q_{1x}) = q_{1x} + s vq_x, \\ y_q(s) &= q_{1y} + s(q_{2y} - q_{1y}) = q_{1y} + s vq_y, \\ z_q(s) &= q_{1z} + s(q_{2z} - q_{1z}) = q_{1z} + s vq_z. \end{aligned} \quad (3.2)$$

para $s \in [0, 1]$ y $vq = q_2 - q_1$. Por otro lado, notemos que, al rotar alrededor del eje z , la superficie obtenida cumplirá la ecuación $\rho^2(s) = x_q^2(s) + y_q^2(s)$. De la ecuación para z_q en 3.2, dado que $vq_z \neq 0$ por hipótesis, podemos despejar y obtener que $s = (z - q_{1z})/vq_z$. Sustituyendo el valor de t en las expresiones para $x_q(s)$ y $y_q(s)$, obtenemos

$$\begin{aligned} x_q(z) &= q_{1x} + (z - q_{1z}) \frac{vq_x}{vq_z} \\ y_q(z) &= q_{1y} + (z - q_{1z}) \frac{vq_y}{vq_z}. \end{aligned} \quad (3.3)$$

Elevando al cuadrado y sumando, obtenemos que

$$\begin{aligned}
x_q^2(z) + y_q^2(z) &= \left(q_{1x} + (z - q_{1z}) \frac{vq_x}{vq_z} \right)^2 + \left(q_{1y} + (z - q_{1z}) \frac{vq_y}{vq_z} \right)^2 \\
&= \left(q_{1x}^2 + 2q_{1x}(z - q_{1z}) \frac{vq_x}{vq_z} + (z - q_{1z})^2 \frac{vq_x^2}{vq_z^2} \right) \\
&\quad + \left(q_{1y}^2 + 2q_{1y}(z - q_{1z}) \frac{vq_y}{vq_z} + (z - q_{1z})^2 \frac{vq_y^2}{vq_z^2} \right) \\
&= q_{1x}^2 + 2q_{1x}(z - q_{1z}) \frac{vq_x}{vq_z} + (z^2 - 2q_{1z}z + q_{1z}^2) \frac{vq_x^2}{vq_z^2} \\
&\quad + q_{1y}^2 + 2q_{1y}(z - q_{1z}) \frac{vq_y}{vq_z} + (z^2 - 2q_{1z}z + q_{1z}^2) \frac{vq_y^2}{vq_z^2} \\
&= z^2 \left(\frac{vq_x^2 + vq_y^2}{vq_z^2} \right) + 2z \left(\frac{vq_x(q_{1x}vq_z - vq_xq_{1z}) + vq_y(q_{1y}vq_z - vq_yq_{1z})}{vq_z^2} \right) \\
&\quad + \left(q_{1x}^2 - 2q_{1x}q_{1z} \frac{vq_x}{vq_z} + \left(q_{1z} \frac{vq_x}{vq_z} \right)^2 \right) + q_{1y}^2 - 2q_{1y}q_{1z} \frac{vq_y}{vq_z} + \left(q_{1z} \frac{vq_y}{vq_z} \right)^2 \\
&= cz^2 + 2rz + d.
\end{aligned} \tag{3.4}$$

■

Podemos simplificar más la expresión para los coeficientes r y d . Sea $\mathbf{r}\mathbf{q} = \mathbf{q}_1 \times \mathbf{q}_2$. Notemos que

$$\begin{aligned}
q_{1x}vq_z - vq_xq_{1z} &= q_{1x}q_{2z} - q_{1x}q_{1z} - q_{2x}q_{1z} + q_{1x}q_{1z} \\
&= q_{1x}q_{2z} - q_{2x}q_{1z} \\
&= -rq_y.
\end{aligned} \tag{3.5}$$

De forma análoga podemos obtener que $q_{1y}vq_z - vq_yq_{1z} = rq_x$. Entonces para r tenemos:

$$r = \left(\frac{-vq_xrq_y + vq_yrq_x}{vq_z^2} \right). \tag{3.6}$$

Por otro lado, para d , obtenemos

$$\begin{aligned}
d &= \left(q_{1x}^2 - 2q_{1x}q_{1z} \frac{vq_x}{vq_z} + \left(q_{1z} \frac{vq_x}{vq_z} \right)^2 + q_{1y}^2 - 2q_{1y}q_{1z} \frac{vq_y}{vq_z} + \left(q_{1z} \frac{vq_y}{vq_z} \right)^2 \right) \\
&= \left(q_{1x} - q_{1z} \frac{vq_x}{vq_z} \right)^2 + \left(q_{1y} - q_{1z} \frac{vq_y}{vq_z} \right)^2 \\
&= \frac{1}{vq_z^2} \left((q_{1x}vq_z - q_{1z}vq_x)^2 + (q_{1y}vq_z - q_{1z}vq_y)^2 \right) \\
&= \frac{(q_{1x}q_{2z} - q_{2x}q_{1z})^2 + (q_{1y}q_{2z} - q_{2y}q_{1z})^2}{vq_z^2} \\
&= \frac{(-rq_y)^2 + rq_x^2}{vq_z^2} = \frac{rq_x^2 + rq_y^2}{vq_z^2}.
\end{aligned} \tag{3.7}$$

Podemos ahora describir la superficie que describe esta ecuación.

Teorema 3.2. *La superficie de revolución generada por girar un segmento de recta no contenido en un plano paralelo al plano XY sobre el eje z es un hiperboloide circular de un manto o un cilindro circular.*

Demostración. Utilizando el lemma 3.1, sabemos que la ecuación de la superficie está dada por

$$x^2 + y^2 - cz^2 - 2rz - d = 0. \tag{3.8}$$

Debemos de distinguir ahora dos casos.

1. $vq_x = vq_y = 0$

Recordando que $c = (vq_x^2 + vq_y^2)/vq_z^2$, entonces $c = 0$ sí y sólo sí $vq_x = vq_y = 0$ pero, por la expresión 3.6, si $vq_x = vq_y = 0$ entonces $r = 0$. Entonces $c = 0$ y $r = 0$ sí y sólo sí $vq_x = vq_y = 0$. Así, si sucede que $vq_x = vq_y = 0$, entonces la ecuación 3.8 se reduce a $x^2 + y^2 - d = 0$, la cual, ya que $d \geq 0$ por expresarse en la ecuación 3.7 como una suma de cuadrados, representa a un cilindro circular de radio \sqrt{d} .

2. $vq_x \neq 0$ o $vq_y \neq 0$.

Podemos reescribir la ecuación 3.8 en forma matricial como:

$$\frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{p}^T \mathbf{x} - d = 0 \tag{3.9}$$

con

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2c \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} 0 \\ 0 \\ -2r \end{pmatrix}.$$

Ya que $vq_x \neq 0$ o $vq_y \neq 0$, podemos garantizar que $c \neq 0$ y así la matriz \mathbf{H} será invertible. Sabemos que entonces podemos hacer un cambio de variable $\mathbf{x} = \mathbf{x}' - \mathbf{H}^{-1}\mathbf{p}$. Al sustituir en la ecuación cuadrática y utilizar que, al ser \mathbf{H} simétrica, entonces $(\mathbf{H}^{-1})^T = (\mathbf{H}^T)^{-1} = \mathbf{H}^{-1}$, tenemos:

$$\begin{aligned} & \frac{1}{2}(\mathbf{x}' - \mathbf{H}^{-1}\mathbf{p})^T \mathbf{H}(\mathbf{x}' - \mathbf{H}^{-1}\mathbf{p}) + \mathbf{p}^T(\mathbf{x}' - \mathbf{H}^{-1}\mathbf{p}) - d \\ &= \frac{1}{2}(\mathbf{x}'^T \mathbf{H} \mathbf{x}' - \mathbf{x}'^T \mathbf{H} \mathbf{H}^{-1} \mathbf{p} - \mathbf{p}^T (\mathbf{H}^{-1})^T \mathbf{H} \mathbf{x}' + \mathbf{p}^T (\mathbf{H}^{-1})^T \mathbf{H} \mathbf{H}^{-1} \mathbf{p}) \\ & \quad + \mathbf{p}^T \mathbf{x}' - \mathbf{p}^T \mathbf{H}^{-1} \mathbf{p} - d \\ &= \frac{1}{2}(\mathbf{x}'^T \mathbf{H} \mathbf{x}' - \mathbf{x}'^T \mathbf{p} - \mathbf{p}^T \mathbf{x}' + \mathbf{p}^T \mathbf{H}^{-1} \mathbf{p}) + \mathbf{p}^T \mathbf{x}' - \mathbf{p}^T \mathbf{H}^{-1} \mathbf{p} - d \\ & \quad = 0 \end{aligned} \tag{3.10}$$

Recordando que $\mathbf{x}'^T \mathbf{p} - \mathbf{p}^T \mathbf{x}' = \mathbf{x} \cdot \mathbf{p}$, podemos eliminar términos y obtener:

$$\frac{1}{2} \mathbf{x}'^T \mathbf{H} \mathbf{x}' + (-d - \frac{1}{2} \mathbf{p}^T \mathbf{H}^{-1} \mathbf{p}) = 0. \tag{3.11}$$

Explícitamente, sabemos que

$$\mathbf{H}^{-1} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & -1/2c \end{pmatrix},$$

por lo que $\frac{1}{2} \mathbf{p}^T \mathbf{H}^{-1} \mathbf{p} = -r^2/c$. Así, si expandimos la ecuación 3.11, obtenemos

$$x'^2 + y'^2 - cz'^2 + (-d + \frac{r^2}{c}) = 0. \tag{3.12}$$

Sabemos que la ecuación 3.12 representa un hiperboloide circular con centro en \mathbf{p} . Para ver que es un hiperboloide de un manto, notemos

que

$$\begin{aligned}
r^2 - cd &= \frac{1}{vq_z^4} ((rq_y vq_x - rq_x vq_y)^2 - (vq_x^2 + vq_y^2)(rq_x^2 + rq_y^2)) \\
&= \frac{1}{vq_z^4} ((rq_y^2 vq_x^2 - 2rq_y vq_x rq_x vq_y + rq_x^2 vq_y^2) \\
&\quad - (vq_x^2 rq_x^2 + vq_x^2 rq_y^2 + vq_y^2 rq_x^2 + vq_y^2 rq_y^2)) \\
&= \frac{1}{vq_z^4} (vq_x^2 rq_x^2 - 2rq_y vq_x rq_x vq_y + vq_y^2 rq_y^2) \\
&= \frac{1}{vq_z^4} (vq_x rq_x - vq_y rq_y)^2 \geq 0.
\end{aligned} \tag{3.13}$$

Entonces, ya que $c \geq 0$, claramente se cumple que $r^2/c - d = 1/c(r^2 - cd) \geq 0$, por lo que la ecuación 3.12 representa un hiperboloide de un manto. Notemos que para el caso $\mathbf{r}\mathbf{q} = \mathbf{0}$, obtenido cuando el segmento q_1q_2 pasa por el origen, se cumple que $r^2/c - d = 0$ y así la superficie obtenida es un cono, un caso degenerado del hiperboloide de un manto. ■

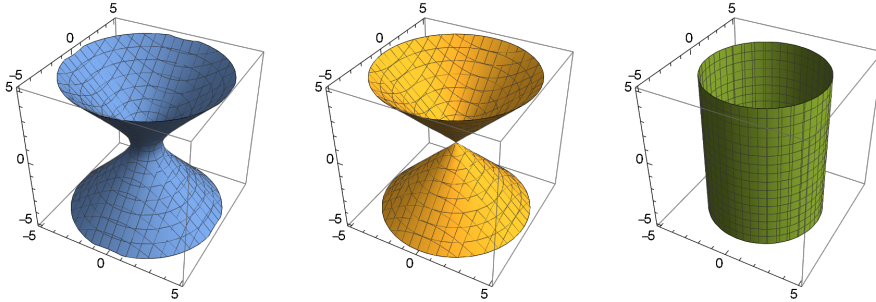


Figura 3.1: Superficies que contienen a las descritas por la rotación de un segmento de recta en \mathbb{R}^3 : hiperboloide de un manto, cono y cilindro.

Ahora que sabemos que la ecuación que describe a la superficie de revolución obtenida, podemos verificar si intersecciona a un segmento arbitrario. Denotamos los extremos de dicho segmento p_1 y p_2 y le damos una parametrización $\mathbf{x}_p = \mathbf{p}_1 + t \mathbf{vp}$ con $t \in [0, 1]$ y $\mathbf{vp} = \mathbf{p}_2 - \mathbf{p}_1$. Si el segmento toca la superficie, eso quiere decir que en algún momento cumple la ecuación 3.8. Al sustituir la expresión parametrizada del segmento p_1p_2 , obtenemos:

$$(p_{1x} + t vp_x)^2 + (p_{1y} + t vp_y)^2 - c(p_{1z} + t vp_z)^2 - 2r(p_{1z} + t vp_z) - d = 0.$$

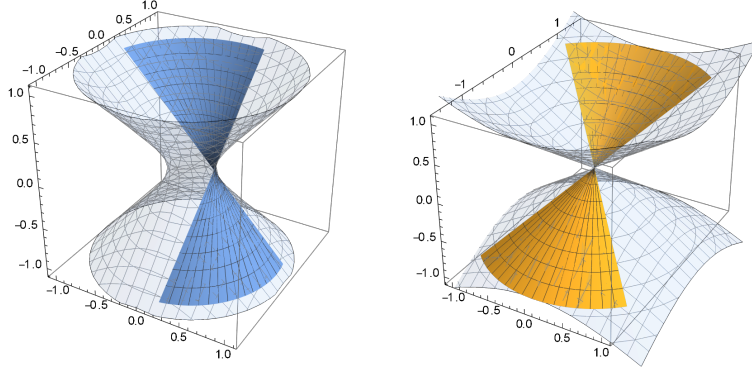


Figura 3.2: Regiones descritas por la rotación de un segmento de recta en \mathbb{R}^3 por un ángulo menor a 2π . Las regiones están contenidas en un hiperboloide y en un cono, respectivamente.

Agrupando los términos respecto a t , se obtiene:

$$t^2 (vp_x^2 + vp_y^2 - cvp_z^2) + 2t (p_{1x}vp_x + p_{1y}vp_y - vp_z(cp_{1z} + r)) + (p_{1x}^2 + p_{1y}^2 - cp_{1z}^2 - 2rp_{1z} - d) = 0. \quad (3.14)$$

La ecuación 3.14 es una ecuación cuadrática que puede resolverse con la fórmula general. Si la ecuación no tiene raíces, entonces no hay intersección. Si la ecuación tiene al menos una raíz y dicha raíz se encuentra en el intervalo $[0, 1]$, entonces existe una intersección entre el segmento y toda la superficie de revolución asociada a la ecuación 3.8.

Sin embargo, recordemos que la ecuación 3.8 describe una superficie de revolución que surge de rotar completamente (es decir, un ángulo 2π) a lo largo de la recta que contiene al segmento q_1q_2 . A nosotros nos interesa solamente la región debida a rotar un ángulo α el segmento. Así, aunque las raíces de la ecuación 3.14 existan y estén en el intervalo deseado, debemos verificar que dicha intersección se de en la región deseada de la superficie.

Notemos que la región debe de tener coordenada $z \in [\text{mín}\{q_{1z}, q_{2z}\}, \text{máx}\{q_{1z}, q_{2z}\}]$. Así, dada una raíz t^* de la ecuación 3.14, solamente si el valor de $z(t^*)$ de la parametrización del segmento p_1p_2 está contenido en el intervalo mostrado, puede existir la intersección entre el segmento y la región.

Sin embargo, debemos ir todavía más lejos pues aunque esté en el intervalo de z correcto, debemos verificar que se da en una región asociada a

un ángulo α y a un valor de s correcto. Notemos que podemos parametrizar dicha región como $\mathbf{R}_z(\alpha)\mathbf{x}_q(t)$ con $\mathbf{R}_z(\alpha)$ la matriz de rotación sobre el eje z por un ángulo α . Explícitamente, las coordenadas cartesianas tienen la forma:

$$\begin{pmatrix} \cos(\alpha)x_q(s) - \sin(\alpha)y_q(s) \\ \sin(\alpha)x_q(s) + \cos(\alpha)y_q(s) \\ z_q(s) \end{pmatrix} \quad \text{con } \alpha \in [0, \alpha], s \in [0, 1]. \quad (3.15)$$

Si dicha superficie se se intersecta con el segmento p_1p_2 , debe hacerlo en una raíz t^* de la ecuación 3.14 y en unos valores particular α^* y s^* , por lo que se cumplirá la igualad

$$\begin{pmatrix} \cos(\alpha^*)x_q(s^*) - \sin(\alpha^*)y_q(s^*) \\ \sin(\alpha^*)x_q(s^*) + \cos(\alpha^*)y_q(s^*) \\ z_q(s^*) \end{pmatrix} = \begin{pmatrix} x_p(t^*) \\ y_p(t^*) \\ z_p(t^*) \end{pmatrix}. \quad (3.16)$$

La condición $z_q(s^*) = z_p(t^*)$ explícitamente se expresa como

$$q_{1z} + s^* vq_z = p_{1z} + t^* vp_z \implies s^* = \frac{p_{1z} + t^* vp_z - q_{1z}}{vq_z}. \quad (3.17)$$

Así, podemos calcular directamente el valor del parámetro s y, al conocer también t^* , la ecuación 3.16 en las componentes x y y de se convierte en un sistema de dos ecuaciones lineales para $\sin(\alpha^*)$, $\cos(\alpha^*)$, con solución:

$$\begin{aligned} \sin(\alpha^*) &= \frac{x_q(s^*)y_p(t^*) - x_p(t^*)y_q(s^*)}{x_q^2(s^*) + y_q^2(s^*)}, \\ \cos(\alpha^*) &= \frac{x_q(s^*)x_p(t^*) + y_p(t^*)y_q(s^*)}{x_q^2(s^*) + y_q^2(s^*)}. \end{aligned} \quad (3.18)$$

α^* se puede calcular partir de estos valores. Ambos parámetros, α^* y s^* , describen el punto de la región en el que se da la intersección con el segmento p_1p_2 . Si dichos parámetros están en el intervalo $[0, \alpha]$ y $[0, 1]$, respectivamente, entonces es seguro que la intersección se da en la región deseada. Este proceso también funciona para el caso en el que nuestro ángulo α es negativo. El único cambio a realizar es que el intervalo para α^* se convierte en $[\alpha, 0]$.

Un caso que también puede surgir es el caso en que los coeficientes de la ecuación 3.14 sean 0. Eso indicaría que la intersección entre la recta que contiene a p_1p_2 y la superficie de revolución asociada a la recta q_1q_2 se da en

una cantidad infinita de puntos. Para verificar que eso se dé en los segmentos y los ángulos deseados, hay que encontrar α , que en este caso es solamente al ángulo entre las proyecciones en el plano XY de las rectas (o entre los puntos vectores p_1 y q_1 , si es que $vq_z = 0$), ver que esté en el intervalo deseado, y luego verificar que el segmento q_1q_2 rotado un ángulo α se encime con el segmento p_1p_2 .

A partir de toda esta información, construimos el algoritmo 1 el cual, dados los extremos p_1, p_2, q_1, q_2 y un ángulo α , nos indique si existe la intersección entre la región generada por rotar el segmento q_1q_2 un ángulo α y el segmento p_1p_2 .

Algoritmo 1 Intersección en el caso 1

```

1: function INTCASO1( $p_1, p_2, q_1, q_2, \alpha$ )
2:   Calcular  $c, r$  y  $d$  como en las ecuaciones 3.4, 3.6, 3.7
3:   if Existen raíces  $t_1, t_2$  de la ecuación 3.14 then
4:     for  $t^* = t_1, t_2$  do
5:       if  $0 \leq t^* \leq 1$  then
6:         if  $z(t^*) \in [\text{mín}\{q_{1z}, q_{2z}\}, \text{máx}\{q_{1z}, q_{2z}\}]$  then
7:           Calcular los parámetros  $s^*, \alpha^*$ 
8:           if  $s^* \in [0, 1]$  &  $\alpha^* \in [0, \alpha]$  then
9:             return True
10:          end if
11:         end if
12:       end if
13:     end for
14:   else if Hay una infinidad de soluciones a la ecuación 3.14 then
15:     Calcular  $\alpha$  como el ángulo tre las proyecciones  $XY$  de  $p_1p_2$  y  $q_1q_2$ 
16:     if  $\alpha \in [0, \alpha]$  then
17:       Sea  $q'_1q'_2$  el segmento obtenido al rotar  $q_1q_2$  un ángulo  $\alpha$ 
18:       if  $q'_1q'_2$  se empalma con  $p_1p_2$  then
19:         return True
20:       end if
21:     end if
22:   end if
23:   return False
24: end function

```

3.2.2. Caso 2: el segmento que rota está en un plano paralelo al eje XY

En este caso, la superficie descrita por el segmento que rota está totalmente contenida en un plano. Es posible describir analíticamente la superficie generada.

Teorema 3.3. *La superficie generada por girar un segmento de recta contenida en un plano paralelo al eje XY está contenida entre dos círculos con centro en el eje z a la altura de dicho plano.*

Demostración. Denotemos los extremos del segmento q_1q_2 y usemos la parametrización usual $\mathbf{x}_q(t) = \mathbf{q}_1 + t \mathbf{v}_q$. Sea $r_1 = \min\{\|\mathbf{x}_q(t)\| \mid t \in [0, 1]\}$ y $r_2 = \max\{\|\mathbf{x}_q(t)\| \mid t \in [0, 1]\}$. Sabemos que r_1, r_2 están bien definidos ya que el segmento es un conjunto compacto.

Rotar el segmento un ángulo α es equivalente a multiplicar por una matriz de rotación sobre el eje z a $\mathbf{x}_q(t)$. Sabemos que ésta es una transformación rígida, por lo que necesariamente se cumplirá $\|\mathbf{R}_z(\alpha)\mathbf{x}_q(t)\| = \|\mathbf{x}_q(t)\|$. Así, es claro que para todo $t \in [0, 1]$, $r_1 \leq \|\mathbf{R}_z(\alpha)\mathbf{x}_q(t)\| \leq r_2$.

Ya que el segmento está contenido en un plano paralelo al XY , sabemos que $v_{q_z} = 0$ y que $z(t) = q_{1z} = q_{2z}$ para todo $t \in [0, 1]$. Ya que la rotación no afecta el eje z , sabemos que $\mathbf{e}_z \cdot \mathbf{R}_z(\alpha)\mathbf{x}(t) = q_{1z}$. Así, dado $C(r) = \{(x, y, z) \in \mathbb{R}^3 \mid z = q_{1z}, x^2 + y^2 \leq r\}$ el círculo de radio r en el plano $z = q_{1z}$, entonces se cumplirá

$$\{\mathbf{R}_z(\alpha)\mathbf{x}(t) \mid \alpha \in [0, 2\pi], t \in [0, 1]\} \subseteq C(r_2) \setminus C(r_1). \quad (3.19)$$

■

Notemos que solamente cuando el segmento se encuentra contenido en una recta que pasa por el eje z entonces los valores de r_1 y r_2 se obtienen en sus extremos y entonces la región obtenida al rotar se puede parametrizar fácilmente. Sin embargo, en general no sucede esto y la región no es sencillamente parametrizable. La figura 3.3 ilustra estas situaciones.

Para verificar dicha región se intersecta con otro segmento de recta p_1p_2 , descrito con la parametrización estándar, lo primero que debemos de checar es que el segmento pase por el plano en el que está contenida la región. Matemáticamente, debemos encontrar el valor del parámetro al que

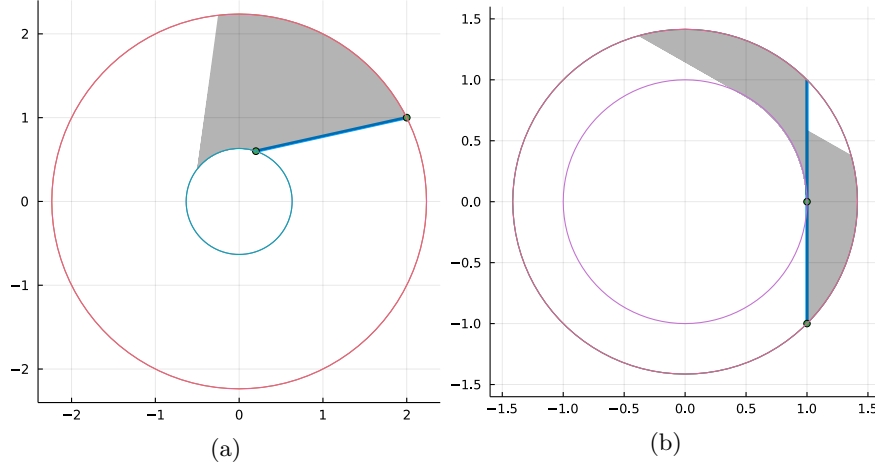


Figura 3.3: Distintas configuraciones para la región generada al rotar una recta contenida en un plano paralelo al XY . En (a) los puntos sobre el segmento más cerca y lejos del origen del origen son los extremos, mientras que en (b) el punto más cercano no corresponde a ningún extremo.

intersecta dicho plano. Usando las ecuaciones para la componente z del segmento estático, esto se expresa como:

$$p_{1z} + t^* vp_z = q_{1z} \implies t^* = \frac{q_{1z} - p_{1z}}{vp_z}. \quad (3.20)$$

Si $t^* \in [0, 1]$ entonces el segmento pasará por dicho plano y podrá existir la intersección. Una condición equivalente para esto también es que $q_{1z} \in [\text{mín}\{p_{1z}, p_{2z}\}, \text{máx}\{p_{1z}, p_{2z}\}]$.

Si alguna de estas condiciones se cumplen, notemos que si la intersección entre los puntos se da, entonces se debe de cumplir la condición $\|\mathbf{R}_z(\alpha)\mathbf{x}_q(s)\| = \|\mathbf{x}_p(t^*)\|$, pero, ya que $\|\mathbf{R}_z(\alpha)\mathbf{x}_q(s)\| = \|\mathbf{x}_q(s)\|$ y ya que $z_q(s) = x_q(t^*) = q_{1z}$, entonces la condición se puede expresar como:

$$x_q(s)^2 + y_q(s)^2 = x_p(t^*)^2 + y_p(t^*)^2. \quad (3.21)$$

Como ya conocemos el valor de t^* , $x_p(t^*)^2 + y_p(t^*)^2 = r^{*2}$ es conocido y si sustituimos las expresiones para $x_q(s)$, $y_q(s)$ obtenemos:

$$(q_{1x} + svp_x)^2 + (q_{1y} + svp_y)^2 = r^{*2}. \quad (3.22)$$

Reagrupando, obtenemos:

$$s^2(vp_x^2 + vp_y^2) + 2s(q_{1x}vp_x + q_{1y}vp_y) + (q_{1x}^2 + q_{1y}^2 - r^{*2}) = 0. \quad (3.23)$$

La ecuación 3.23 es una ecuación cuadrática para s . Si tiene una raíz s^* tal que $s^* \in [0, 1]$, entonces la intersección entre los segmentos podría existir. Para garantizar que dicha intersección realmente exista, solo debemos encontrar el valor del parámetro α^* que cumpla las ecuaciones 3.16 para los valores ya encontrados de s^* y t^* y ver que cumpla $\alpha^* \in [0, \alpha]$.

Con base en todo lo anterior, construimos el algoritmo 2 para decidir si una intersección se da en el caso descrito. El algoritmo en este caso resulta mucho más sencillo pues en este caso resulta imposible que la intersección se de en una cantidad infinita de puntos.

Algoritmo 2 Intersección en el caso 2

```

1: function INTCASO2( $p_1, p_2, q_1, q_2, \alpha$ )
2:   if  $q_{1z} \in [\text{mín}\{p_{1z}, p_{2z}\}, \text{máx}\{p_{1z}, p_{2z}\}]$  then
3:     Calcular  $t^*$  de la ecuación 3.20
4:     Calcular las raíces  $s_1, s_2$  de la ecuación 3.23
5:     for  $s^* = s_1, s_2$  do
6:       if  $0 \leq s^* \leq 1$  then
7:         Calcular  $\alpha^*$ 
8:         if  $\alpha^* \in [0, \alpha]$  then
9:           return True
10:        end if
11:       end if
12:     end for
13:     return False
14:   end if
15: end function

```

3.2.3. Caso 3: Ambos segmentos están en un plano paralelo al eje XY

En este caso, sabemos que la región descrita por la rotación del segmento q_1q_2 será igual que en el caso anterior. Sin embargo, la verificación de la intersección cambia de forma dramática, pues, a diferencia de los casos anteriores, si existe una intersección entre la región el segmento que gira, esta consistirá en una cantidad infinita de puntos.

Sin embargo, podemos evadir este problema de la siguiente manera: sabemos, por el teorema 3.3, que la región generada por el segmento que gira estará contenida entre dos círculos de radio r_1, r_2 que son las distancias

máximas y mínimas, respectivamente, del segmento que rota al origen. Si dicha región intersecta a otro segmento contenido en ese plano, independientemente de la cantidad de puntos en los que se de dicha intersección, al menos uno de ellos debe estar en una circunferencia con centro en el origen que pase por alguno de q_1, q_2, p_1 o p_2 .

Lema 3.4. *Sea q_1q_2 un segmento en \mathbb{R}^2 y α un ángulo dado. Entonces la distancia mínima o máxima del segmento a la recta se alcanza en q_1 o en q_2 .*

Demostración. Notemos que la distancia al cuadrado de un punto sobre la recta que pasa por q_1q_2 tiene la forma:

$$f(t) = x_q^2(t) + y_q^2(t) = (p_{1x} + t vp_{1x})^2 + (p_{1y} + t vp_{1y})^2. \quad (3.24)$$

Si derivamos respecto a t e igualamos a 0, podemos encontrar el valor t' cuando se da la distancia mínima

$$\begin{aligned} \left. \frac{df}{dt} \right|_{t=t'} &= 2(p_{1x} + t' vp_{1x})vp_{1x} + 2(p_{1y} + t' vp_{1y})vp_{1y} = 2(\mathbf{p}_1 + t'\mathbf{vp}) \cdot \mathbf{vp} = 0 \\ &\implies t' = -\frac{\mathbf{p}_1 \cdot \mathbf{vp}}{\mathbf{vp} \cdot \mathbf{vp}}. \end{aligned} \quad (3.25)$$

Por la desigualdad del triángulo, podemos garantizar que los puntos a los cuales se dan las distancias mínimas y máximas del segmento q_1q_2 al origen respectivamente serán:

$$\begin{cases} \mathbf{q}_1, \mathbf{q}_2 & \text{si } t' < 0 \\ \mathbf{x}_q(t'), \mathbf{q}_2 & \text{si } 0 \leq t' < 0.5 \\ \mathbf{x}_q(t'), \mathbf{q}_1 & \text{si } 0.5 \leq t' < 1 \\ \mathbf{q}_2, \mathbf{q}_1 & \text{si } t' \geq 1 \end{cases} \quad (3.26)$$

■

Teorema 3.5. *En \mathbb{R}^2 , sea un segmento p_1p_2 y una región R obtenida por girar un ángulo α un segmento q_1q_2 . Entonces R se intersecta con p_1p_2 sí y sólo sí sucede alguno de los siguiente casos:*

- Los segmentos p_1p_2 y q_1q_2 ya se intersectan.*
- Existe un punto w en la circunferencia centrada en el origen que pasa por $u \in \{q_1, q_2\}$ tal que $w \in R \cap p_1p_2$ y el ángulo interno entre el vector \mathbf{w} y \mathbf{u} está en el intervalo $[0, \alpha]$.*

- c) Existe un punto w en la circunferencia centrada en el origen que pasa por $u \in \{p_1, p_2\}$ tal que $w \in R \cap p_1p_2$ y el ángulo interno entre el vector \mathbf{w} y \mathbf{u} está en el intervalo $[-\alpha, 0]$.

Demostración. Las implicaciones en dirección \Leftarrow se siguen por definición, por lo que solo nos encargamos de probar \Rightarrow . Para eso, supongamos que tenemos los segmentos y un ángulo α tal que rotar q_1q_2 genera la intersección mencionada. Notemos, por la figura 3.3, que la frontera de la región R siempre contiene a los arcos de circunferencia resultados de girar \mathbf{q}_1 y \mathbf{q}_2 un ángulo α . Más aún, las partes de la frontera que no son dichos arcos de circunferencia, son segmentos de recta o son un arco de circunferencia asociado a la rotación del punto d sobre q_1q_2 más cercano al origen.

Por la forma en que está definida R , notemos que su frontera ∂R es una curva simple siempre y cuando $\alpha \in [-\pi, \pi]$. Así, por el teorema de la curva de Jordan aplicado a ∂R , sabemos que el plano se particiona totalmente entre R y su complemento $\mathbb{R}^2 \setminus R$ de tal forma que la intersección entre p_1p_2 y R se da tal que hay al menos un punto en $p_1p_2 \cap \partial R$ y otro punto en $p_1p_2 \cap \mathbb{R}^2 \setminus R$ o con p_1p_2 completamente contenido en R . Ambos casos se muestran en la figura 3.4. Si el segmento está totalmente contenido en R ,

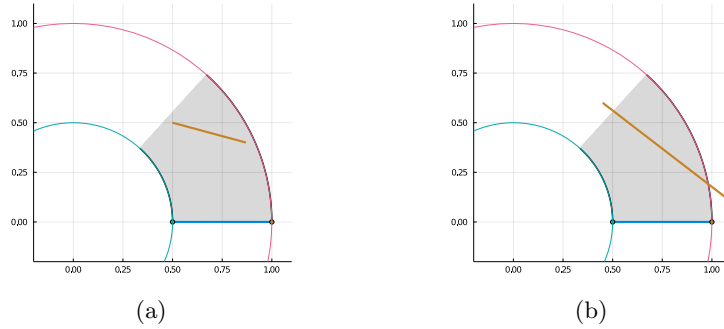


Figura 3.4: Posibilidades de la intersección respecto a la frontera entre un segmento de recta y la región descrita por otro segmento que rota en el mismo plano. (a) El segmento está totalmente contenido en la región. (b) Hay un pedazo contenido y otro fuera.

por la parametrización de R , entonces existe un $\alpha_1, \alpha_2 \in [0, \alpha]$ y $t_1, t_2 \in [0, 1]$ tal que $\mathbf{R}_z(\alpha_1)\mathbf{x}_q(t_1) = \mathbf{p}_1$ o $\mathbf{R}_z(\alpha_2)\mathbf{x}_q(t_2) = \mathbf{p}_2$. Así, si ahora consideramos que $-\alpha^* \in [-\alpha, 0]$, entonces se cumplirá que $\mathbf{x}_q(t_1) = \mathbf{R}_z(-\alpha_1)\mathbf{p}_1$ como $\mathbf{x}_q(t_2) = \mathbf{R}_z(-\alpha_2)\mathbf{p}_2$. Eso significa que al girar el segmento p_1p_2 un ángulo $-\alpha_1$, el arco trazado por p_1 toca a la recta q_1q_2 . Entonces se cumple b)

Si se cumple que hay un $u \in p_1p_2 \cap \partial R \subset p_1p_2 \cap R$, entonces, como los arcos generados por q_1, q_2 , están en la frontera, hay dos casos:

1. Si el punto u interseca a la frontera en alguno de dichos arcos, entonces se cumple c). Esto se ve en la figura 3.4(b).
2. Si el punto u no interseca a la frontera alguno de dichos arcos, entonces lo hace en alguna de las otras curvas. Ya que p_1p_2 no está totalmente contenido en el interior de R , entonces garantizamos que hay un pedazo de dicho segmento fuera de R . Debemos ahora tomar casos sobre la cardinalidad de $\partial R \cap p_1p_2$:

a) $|\partial R \cap p_1p_2| = 1$ (figura 3.5(a)).

Coloquialmente, esto quiere decir que un pedazo del segmento está dentro de R y sale de R por una parte de la frontera que no es un arco de circunferencia de los extremos de q_1q_2 . Es decir, en algún ángulo alguno de los extremos p_1 o p_2 toca a la frontera de R para empezar a introducirse en él.

Sea $u \in \partial R \cap p_1p_2$. Sin pérdida de generalidad, podemos suponer que el segmento up_1 está totalmente contenido en R , por lo que, al igual que en el caso en el que el segmento está totalmente contenido, entonces se cumplirá b).

b) $|\partial R \cap p_1p_2| \geq 2$.

En este caso, ya que sabemos, por el teorema 3.3, que la región R está contenida entre dos circunferencias, se debe cumplir alguno de los siguientes casos:

- 1) $|\partial R \cap p_1p_2| \geq 2$ y al menos dos intersecciones se dan con el arco generado al rotar d (figura 3.5(b)).

Podemos encontrar un subsegmento $v_1v_2 \subset p_1p_2$ tal que $|\partial R \cap v_1v_2| = 1$ y se cumplan las condiciones del caso $|\partial R \cap v_1v_2| = 1$, por lo que podremos garantizar que se cumple c)

- II) $|\partial R \cap p_1 p_2| = 2$ y una intersección es con un arco de d , otra con una recta (figura 3.5(c)).

Notemos que la recta con la que se da la intersección no puede intersectarse con el arco obtenido al rotar d , por lo que debe ser una recta correspondiente al segmento $q_1 q_2$ sin rotarse o que toca a dicho segmento desde el lado contrario al que está el arco por girar d . Al estar en el lado contrario respecto a $q_1 q_2$, entonces podemos garantizar que el $p_1 p_2$ debe intersectar a $q_1 q_2$, por lo que se cumple *a*).

- III) $|\partial R \cap p_1 p_2| = 2$ y ninguna intersección se da con los arcos de d (figura 3.5(d)).

En este caso, la única alternativa es que las dos intersecciones se den con segmentos de recta en la frontera. Necesariamente, uno de esos dos segmentos es parte de la recta estática antes de rotarla. Así, los segmentos ya se intersectaban antes de rotar y se cumple el caso *a*.

■

Podemos usar como base este teorema para verificar si se da una intersección en este caso o no. Notemos que la circunferencia que pasa por un punto p centrado el origen tiene radio $|p|$, por lo que, dado q y otro segmento $q_1 q_2$, podemos encontrar la intersección entre la recta que contiene a $q_1 q_2$ y la circunferencia usando la ecuación 3.23. El algoritmo 3 permite hacer dicha verificación. Notemos que el caso en el que el segmento estático está contenido en un plano paralelo al XY ya está cubierto por el caso 1. Así, con estos tres casos cubiertos, podemos construir un algoritmo general que nos permita verificar si al rotar un segmento de recta se da una intersección con otro segmento de recta.

3.2.4. Decidibilidad de modificar una cadena rotando un ángulo diedro

Dada una cadena poligonal $P = (\mathbf{v}_0, \dots, \mathbf{v}_n)$ de longitud n , sabemos que tendrá $n - 1$ ángulos internos y $n - 2$ ángulos diedros. Si tomamos un índice arbitrario $k \in \{1, \dots, n - 2\}$ y un ángulo $\phi \in [-\pi/4, \pi/4]$, quisiéramos

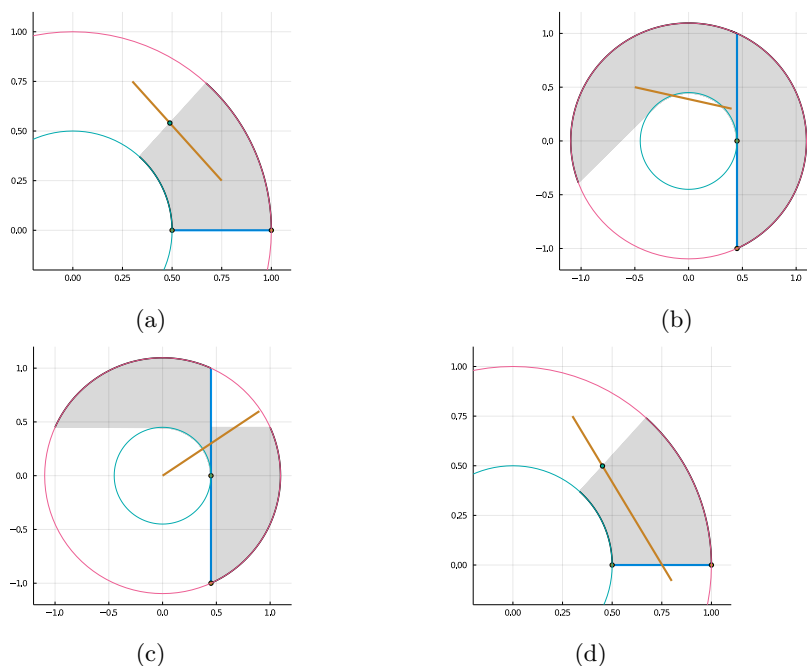


Figura 3.5: Subcasos de posibles intersecciones para el caso 3 cuando el segmento no está completamente contenido en la región.

decidir si es posible rotar el i -ésimo ángulo diedro de la cadena poligonal un ángulo ϕ sin generar ninguna intersección entre los segmentos.

Recordando lo mencionado al inicio de la sección 3.1, cambiar el k -ésimo ángulo diedro por un valor ϕ es equivalente a rotar todos los vértices \mathbf{v}_i para $k+2 \leq i \leq n$ respecto al plano ortogonal al vector $\mathbf{v}_{k+1} - \mathbf{v}_k$ un ángulo ϕ , mientras dejamos intactos los vértices \mathbf{v}_j con $0 \leq j \leq k+1$ intactos. Así, verificar si es posible cambiar el k -ésimo ángulo diedro de un sistema por un valor ϕ es equivalente a revisar si se da alguna intersección al rotar el segmento $\mathbf{v}_{i-1}\mathbf{v}_i$ con algún segmento $\mathbf{v}_{j-1}\mathbf{v}_j$. Notemos que cuando $i = k+2$, $j = k+1$ los segmentos se intersectan en el vértice \mathbf{v}_{k+1} , por lo que ese caso debe ser evitado.

Así, al ya contar con el algoritmo 4, podemos desarrollar otro algoritmo para verificar si es posible cambiar el ángulo diedro de un sistema sin generar ninguna intersección al verificar estas intersecciones entre pares de segmentos. Notemos que el algoritmo 5 tiene una complejidad temporal $\mathcal{O}(n^2)$ y espacial $\mathcal{O}(n)$. El enfoque se hace directamente en tres dimensiones, aunque

Algoritmo 3 Intersección en el caso 3

```

1: function INTCASO3( $p_1, p_2, q_1, q_2, \alpha$ )
2:   for  $q = q_1, q_2$  do
3:     Encontrar el parámetro  $t^*$  de la intersección entre la circunferencia centrada en el origen que pasa por  $q$  y la recta que contiene  $p_1 p_2$  usando la ecuación 3.23
4:     if  $t^* \in [0, 1]$  then
5:       Encontrar el ángulo  $\alpha^*$  entre  $\mathbf{x}_p(\mathbf{t}^*)$  y  $q$ 
6:       if  $\alpha^* \in [0, \alpha]$  then
7:         return True
8:       end if
9:     end if
10:  end for
11:  for  $p = p_1, p_2$  do
12:    Encontrar el parámetro  $s^*$  de la intersección entre la circunferencia centrada en el origen que pasa por  $p$  y la recta que contiene  $q_1 q_2$  usando la ecuación 3.23
13:    if  $s^* \in [0, 1]$  then
14:      Encontrar el ángulo  $\alpha^*$  entre  $\mathbf{x}_q(\mathbf{s}^*)$  y  $p$ 
15:      if  $\alpha^* \in [-\alpha, 0]$  then
16:        return True
17:      end if
18:    end if
19:  end for
20:  return False
21: end function

```

es posible reducir el problema a verificar la intersección entre un conjunto de arcos hiperbólicos en un plano [43]. Sin embargo, aunque esta reducción puede resolverse con complejidad $\mathcal{O}(n \log(n) + k)$ utilizando el algoritmo de Balaban para intersecciones de segmentos [44], el aumento no resulta tan significativo dado que $k \in \mathcal{O}(n^2)$. Es posible demostrar también que la complejidad temporal del problema también cumple $\Omega(n \log n)$.

El caso $\phi = 2\pi$ sí puede manejarse con una complejidad $\mathcal{O}(f^{-1}(n)n \log n)$ con $f(n, n) = n$ la función de Ackermann [43]. Sin embargo, dicho caso resulta de poca utilidad a la hora de trabajar con cadenas poligonales que presenten nudos que ya eviten la posibilidad de cambiar un ángulo diedro 2π . Más aún, nos interesa más el caso en el cual $\phi \approx 0$, por lo que es poco

Algoritmo 4 Intersección general segmento - segmento rotante

```

1: function INTSEGMENTO( $p_1, p_2, q_1, q_2, \alpha$ )
2:   Sea  $vq = q_2 - q_1, vp = p_2 - p_1$ 
3:   if  $vq_z \not\approx 0$  then
4:     return intCaso1( $p_1, p_2, q_1, q_2, \alpha$ )
5:   else if  $vq_z \approx 0 \wedge vp_z \not\approx 0$  then
6:     return intCaso2( $p_1, p_2, q_1, q_2, \alpha$ )
7:   else
8:     return intCaso3( $p_1, p_2, q_1, q_2, \alpha$ )
9:   end if
10: end function

```

Algoritmo 5 Posibilidad de cambiar un ángulo diedro

```

1: function CHECARCAMBIARDIEDRO( $P, k, \phi$ )
2:   Rotar la y trasladar la cadena  $P$  tal que el vector  $\mathbf{u}_k$  sea colineal a
    $\mathbf{e}_z$ 
3:   Sea problema = False
4:   sea  $i = k + 2$ 
5:   while  $i \leq n \wedge \neg$ problema do
6:     while  $i \leq k + 1 \wedge \neg$ problema do
7:       if  $j \neq k + 1 \wedge i \neq k + 2$  then
8:         Sea interseccion = intSegmento( $\mathbf{v}_{j-1}, \mathbf{v}_j, \mathbf{v}_{i-1}, \mathbf{v}_i, \phi$ )
9:         Sea problema = problema  $\vee$  interseccion
10:      end if
11:       $i = i + 1$ 
12:       $j = j + 1$ 
13:    end while
14:  end while
15:  return problema
16: end function

```

realista el caso $\phi = 2\pi$.

De forma análoga, podemos construir el algoritmo para verificar si es posible modificar el k -ésimo por un valor ϕ , simplemente debemos de cambiar el tipo de movimiento rígido inicial de la cadena al movimiento que pone al vector \mathbf{n}_k en la dirección de \mathbf{e}_z y cambiar los límites de los segmentos entre los que se busca intersección.

A la hora de la implementación, es necesario el uso de aritmética de precisión arbitraria [45, Capítulo-4.3] para lograr detectar las intersecciones en los casos más complicados. Ya que no esto no es central a nuestro trabajo, nos permitimos el uso de una de las múltiples implementaciones ya existentes para la aritmética de precisión arbitraria [46].

3.3. Algoritmos para optimización

Pretendemos llevar nuestra cadena desde una configuración dada hasta otra configuración canónica mediante un proceso de optimización numérica. Remitimos al lector a [47, 48] para una introducción al tema.

Para poder llevar nuestra cadena desde una configuración inicial hasta una configuración que minimice globalmente la energía seleccionada para la cadena, es necesario utilizar un algoritmo de minimización que sea capaz de realizar minimización global, diferente a los algoritmos convencionales basados en gradientes que solo nos permiten realizar minimización local. Sin embargo, al intentar utilizar optimización global nos enfrentamos al problema de que, aunque el algoritmo seleccionado pueda tener una demostración de convergencia hacia un mínimo global bajo ciertas condiciones de regularidad, es muy probable que en la práctica sea muy complicado conseguir optimizar de forma global.

En general, trataremos con algoritmos que en la literatura son agrupados bajo el nombre de **metaheurísticas**. Las heurísticas pueden definirse como un algoritmo o método para resolver un problema particular y bien definido, cuya justificación es más bien empírica que teórica, es decir, que en la práctica funciona aunque no haya un formalismo sólido que explique dicho comportamiento. Así, una metaheurística puede entenderse como una heurística que funciona no para un problema en particular pero para una clase más grande de problemas.

3.3.1. Recocido simulado

El **recocido simulado** (*simulated annealing*) es un algoritmo de optimización inspirado en el proceso físico de recocido. El recocido de un material es una técnica de cristalización en la cual un material a cristalizarse primero es calentado hasta temperaturas muy altas. Después, el material es enfriado de forma lenta y controlada hasta muy bajas temperaturas. Este proceso permite que haya un gran movimiento en la escala atómica en las altas tem-

peraturas y, al enfriarlos, los átomos comiencen a reducir su movimiento y la gran mayoría acomodan en posiciones que corresponden al mínimo global de la energía del material ¹. Esto a su vez está basado en la tercera ley de la termodinámica, que indica que conforme la temperatura se va a 0, la configuración de un sistema tiende a la que minimiza globalmente su energía libre.

En la física estadística, las propiedades termodinámicas de una sistema se pueden simular mediante el algoritmo de Metropolis-Hastings [49, 50], un método de Monte Carlo basado en procesos de Markov que nos permite muestrear la distribución de configuraciones del material en equilibrio termodinámico a una temperatura dada. Este es un método estándar para resolver muchos sistemas ya que generalmente es imposible conocer de forma analítica la distribución de equilibrio de un material. Así, tomando como base este algoritmo, es posible emular el proceso de recocido en un material: dada una temperatura inicial, primero utilizamos Metropolis-Hastings para simular el material y, después de iteraciones que garanticen su convergencia al equilibrio termodinámico, disminuimos su temperatura. Repetimos este proceso hasta que la temperatura sea menor que el valor deseado.

Podemos convertir directamente este método en un algoritmo de optimización: lo que corresponde a la energía, en este caso, es una función arbitraria $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$. El espacio de configuraciones consiste en el dominio de la función X y una configuración del material es simplemente un punto $x \in X$. La temperatura $T \in \mathbb{R}^+$ es simplemente un parámetro controlado por el algoritmo. A partir de un punto inicial x , generaremos una nueva configuración x' y la aceptaremos con la regla de Metropolis-Hastings, según la cuál aceptamos x' con probabilidad

$$p(x, x', T) = \min \left\{ \exp \left(\frac{-f(x') + f(x)}{T} \right), 1 \right\}. \quad (3.27)$$

La regla de aceptación nos da mucha información relevante sobre el algoritmo: en primer lugar, notemos que si $f(x') < f(x)$, entonces, ya que $T > 0$, $\exp((-f(x') + f(x))/T) > 1$. Esto quiere decir que siempre aceptamos los pasos hacia configuraciones donde la energía sea menor. Por otro lado, si $f(x') > f(x)$, entonces solo aceptaremos el paso con probabilidad $\exp((-f(x') + f(x))/T) < 1$. Si estamos en este caso y además y tenemos

¹La configuración atómica que comúnmente minimiza la energía de un sistema así es una estructura cristalina: los átomos se organizan en una teselación regular del espacio que tiene simetría traslacional y rotacional

$T_1 < T_2$, entonces $(-f(x') + f(x))/T_2 > (-f(x') + f(x))/T_1$, por lo que $p(x, x', T_2) > p(x, x', T_1)$: la probabilidad de ir hacia una configuración que aumente la energía es mayor a temperaturas altas y decrece conforme disminuimos la temperatura, como lo muestra la figura 3.6.

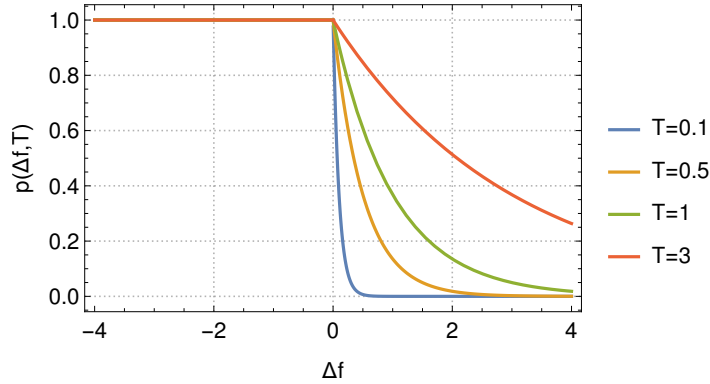


Figura 3.6: Probabilidad de aceptación de un cambio de configuración como función de $\Delta f = f(x') - f(x)$ para distintas temperaturas.

La propiedad discutida en el último paso es crucial para que este algoritmo nos permita minimizar de forma global ya que permite que, a diferencia de minimizaciones basadas en gradientes, no nos movamos de forma continua al mínimo local más cercano sino que exploremos el espacio de configuraciones en búsqueda de mínimos y, conforme disminuimos la temperatura, comencemos a dejar de explorar el espacio y a acercarnos al mínimo cercano a nosotros, el cual probablemente sea el mínimo global si realizamos una exploración suficiente del espacio.

El algoritmo 6 esboza el recocido simulado para la minimización de una función $f(x)$. Notemos que, aunque nosotros planeamos el algoritmo para el caso de funciones reales, es suficientemente general para tratar con problemas de optimización combinatoria o con cualquier otro dominio. Existen diversas reglas para actualizar T que se pueden encontrar en la literatura [51]. Enlistamos las más comunes:

Geométrico $T_{k+1} = rT_k$ con $r \in [0.5, 1)$. Generalmente, $r \approx 0.95$

Lineal $T_{k+1} = T_k - \beta$ con $0 \leq \beta \leq T_0$.

Logarítmico $T_k = T^*/(\log(k+1))$ con $T^* = \log(2)T_0$.

Algoritmo 6 Recocido simulado

```

1: function RECOCIDOSIMULADO( $f, x_0, T_0, T_{min}, N_T$ )
2:   Sea  $T = T_0, x = x_0$ 
3:   while  $T \geq T_{min}$  do
4:     for  $i = 1, \dots, N_T$  do
5:       Generar  $x'$  a partir de  $x$ 
6:       Sea  $r \sim U([0, 1])$ 
7:       if  $r < \exp((-f(x') + f(x))/T)$  then
8:         Sea  $x = x'$ 
9:       end if
10:    end for
11:    Reducir  $T$ 
12:  end while
13:  return  $x$ 
14: end function

```

Una ventaja de la regla logarítmica es que se ha demostrado la convergencia al mínimo global al emplear esa regla. También es posible utilizar temperaturas no monótonas pero que tienden a 0, aunque no es común. La figura 3.7

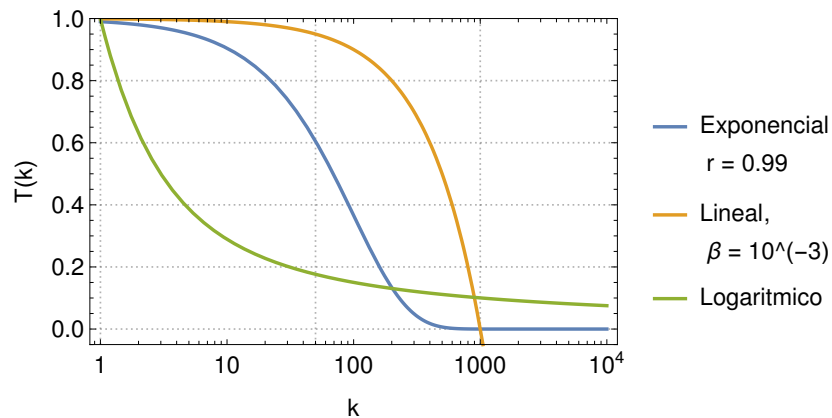


Figura 3.7: Distintos programas de temperatura. La escala del eje x es logarítmica.

La regla para generar nuevas configuraciones x' se realiza de forma aleatoria y distinta entre el tipo de problema que tengamos (optimización real, combinatoria, etc). Para funciones reales, normalmente se toma $x' = x + Z$

con Z un vector aleatorio con una distribución dada (generalmente, una distribución normal con media 0).

Algo importante de notar es que, ya que la generación de configuraciones es aleatoria, el algoritmo se adapta inmediatamente al caso en el que el dominio es no convexo o tiene cualquier tipo de restricción. Si generamos una configuración que se sale del dominio, basta seguir generando configuraciones hasta tener una que está en el dominio. Este es nuestro caso, pues para la cadena poligonal, generamos una nueva configuración moviendo un ángulo y debemos verificar que es posible llegar a dicha configuración sin generar intersecciones.

Las condiciones para la convergencia del algoritmo al mínimo global pueden ser distintas dependiendo del problema que tengamos. Para optimización real, podemos dar condiciones precisas para la convergencia [52]. A pesar de que existan estas condiciones de convergencia, en la práctica hay que jugar con los hiperparámetros del algoritmo. En particular, es muy importante tomar un T_0 suficientemente alto que permita que al inicio se recorra bien el dominio de la función. Para visualizar algunos ejemplos de cómo actúa el recocido simulado sobre algunas funciones $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, el algoritmo se implementó y se probó para tres funciones distintas, dos de ellas con restricciones en el dominio y una con un dominio no convexo. Los resultados de dicho proceso se pueden consultar en el **Anexo A** en el final de este trabajo. En dichos resultados se observa como no siempre se logra minimizar globalmente la función.

3.3.2. Algoritmos Genéticos

Los algoritmos genéticos son otra metaheurística usada para optimizar, inspirada en el proceso biológico de selección natural. Los algoritmos genéticos no parten de una sola posible solución, sino de un conjunto de m posibles soluciones al problema. Llamamos a este conjunto de soluciones la **población** de soluciones y a m el tamaño de la población. Los algoritmos genéticos mejoran de forma iterativa una población de soluciones mediante tres procedimientos básicos:

- **Selección:** para cada elemento de la población, se selecciona uno o más elementos que tomarán el papel de sus padres. El criterio para seleccionar padres debe ser estocástico y darle prioridad a las soluciones con mayor peso.

- **Reproducción:** combinamos entre sí a los padres seleccionados en el paso anterior para formar m nuevas soluciones.
- **Mutación:** modificamos ligeramente las m soluciones obtenidas después de la reproducción. Esto tiene como finalidad alejar a las soluciones de las características que ya poseen las soluciones existentes.

Cada uno de estos procesos puede realizarse de distintas maneras dependiendo del tipo de problema de optimización que tengamos. Se puede argumentar que el proceso de selección es el más importante de todo el algoritmo. Describimos tres formas convencionalmente utilizadas para seleccionar un número arbitrario de padres para cada elemento de la población:

Selección por truncamiento: los padres se escogen de forma aleatoria, independiente e uniforme entre las mejores $k < m$ soluciones de toda la población.

Selección por torneo: cada padre se escoge de forma independiente como el mejor de entre una muestra aleatoria de $k < m$ soluciones.

Selección de ruleta: la solución x_i se escoge como padre con una probabilidad inversamente proporcional a $f(x_i)$.

La figura 3.8 ejemplifica los distintos métodos de selección.

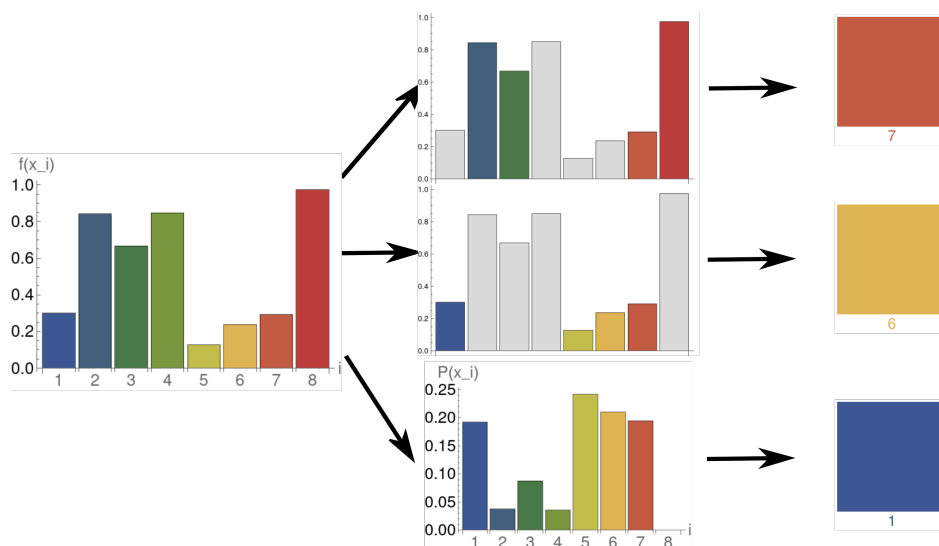


Figura 3.8: Métodos de selección para un algoritmo genético: truncamiento con $k = 4$ (arriba), torneo con $k = 4$ (enmedio) y de ruleta (abajo). En la primera columna, una gráfica de barras con los valores de una función f para una población de tamaño $m = 8$. En la segunda columna, para el truncamiento se muestra en color la muestra aleatoria, para el torneo los mejores k elementos y para la ruleta, los valores de la función transformados en probabilidad. La última columna tiene al individuo seleccionado como padre.

Así, podemos describir brevemente un algoritmo genético en el algoritmo 7. Dependiendo del tipo de problema de optimización que tengamos, la forma de realizar la reproducción y la mutación será distinta. Nuevamente, para el caso de la optimización de una función real $f : \mathbb{R}^n \rightarrow \mathbb{R}$, entonces la forma más usual de reproducir es obtener dos padres para cada solución x_{i1}, x_{i2} y definir $x'_i = x_{i1} + \lambda(x_{i2} - x_{i1})$. Para la mutación, de la misma forma que en el recocido simulado, lo más sencillo es sumar un vector aleatorio Z_i para obtener $x''_i = x'_i + Z_i$ con Z_i un vector aleatorio con una distribución dada (generalmente normal con media 0).

La diferencia principal entre los algoritmos genéticos y el recocido simulado es el hecho de que los algoritmos genéticos permiten que las soluciones intercambien información entre ellas, permitiendo una exploración más informada del espacio de configuraciones. En la práctica, esto se ve reflejado con un menor número necesario de iteraciones para que el algoritmo converja. Existen muchas otras metaheurísticas inspiradas en poblaciones biológicas

Algoritmo 7 Algoritmo genético

```

1: function ALGORITMOGENETICO( $f, x_1, \dots, x_m, \text{tol}$ )
2:   while  $\max\{f(x_i) \mid 1 \leq i \leq m\} > \text{tol}$  do
3:     for  $i = 1, \dots, m$  do
4:       Seleccionar los padres de  $x_i$  usando como referencia  $\{f(x_i) \mid$ 
5:          $1 \leq i \leq m\}$ 
6:       Obtener  $x'_i$  a partir de reproducir a los padres de  $x_i$ 
7:       Obtener  $x''_i$  a partir de mutar a  $x'_i$ 
8:     end for
9:     for  $i = 1, \dots, m$  do
10:      Sea  $x_i = x''_i$ 
11:    end for
12:  end while
13: return  $x$ 
14: end function

```

que se apoyan en el mismo principio de intercambio de información para generar buenos resultados [51].

Sin embargo, a diferencia del recocido simulado, no es tan sencillo utilizar un algoritmo genético para el caso de un dominio no convexo o con restricciones. Aunque en el caso de la mutación podemos simplemente aplicar la misma regla de solo aceptar la mutación de individuo si se queda en el dominio, para la regla de reproducción planteada aquí, el nuevo elemento de la solución solo estará en el dominio si este es convexo. Así, podemos plantear nuevas reglas de reproducción o simplemente evitar dicho paso.

Aunque se ha estudiado la convergencia de los algoritmos genéticos, no tan sencillo dar condiciones para un problema de optimización continua que garanticen la convergencia del algoritmo, aunque dichas condiciones existen [53].

En el **Anexo A** mostramos la forma de actuar de los algoritmos genéticos para algunas funciones de \mathbb{R}^2 .

3.3.3. Descenso de gradiente fraccionario (FGD)

Los métodos de descenso de gradiente son los algoritmos más comúnmente utilizados para minimizar funciones que son derivables y cumplen con ciertas condiciones de regularidad sobre sus derivadas (como ser de clase C^1).

Estos métodos solo nos permiten encontrar mínimos locales de la función.

El descenso de gradiente cuenta con la virtud de que en cada paso del algoritmo, la solución al problema se mueve en la dirección en la que decrece más. Esto siempre resulta más eficiente que buscar en forma aleatoria un lugar en el que la función sea menor. Sin embargo, si la dirección en la que la función decrece saca a la solución del dominio de la función, no hay forma de plantear otra dirección de movimiento.

En principio, es posible aplicar directamente un método de descenso de gradiente a las coordenadas de una cadena poligonal para moverla en la dirección que minimice una energía derivable. Sin embargo, no hay forma de garantizar que dicho movimiento va a mover la cadena en un movimiento que no genere una intersección. Una forma de evitar las intersecciones es utilizar una energía que evite las intersecciones entre la cadena y contar con un paso de aprendizaje suficientemente pequeño para que el movimiento sea pequeño y no haya cambios abruptos en la cadena. Así, la mejor opción para una función de energía es la energía de punto tangente formulada en la ecuación 2.18, pues dicha función evita tanto las intersecciones como el apretamiento de los nudos y tiene pocos mínimos locales [17].

Sin embargo, en la práctica, se puede encontrar que el gradiente tradicional de la energía de punto tangente no logra dar una dirección óptima de desenredamiento. Dicho problema se origina en el hecho de que para minimizar la energía de punto tangente el descenso de gradiente da en cada punto de la curva una dirección basada en cambios locales de la función y es necesario considerar el valor de la función globalmente en la cadena, no solo localmente. Se ha encontrado [18] que es posible obtener una dirección para desenredar una curva utilizando un gradiente fraccionario en lugar del gradiente tradicional. La figure 3.9 muestra la diferencia entre el gradiente tradicional y el fraccionario para una curva suave que presenta un nudo. El fundamento teórico de análisis fraccionario y espacios fraccionarios de Sobolev necesario para deducir el algoritmo escapa al alcance de este trabajo. El lector interesado puede consultarlo en [54, 18]. En primer lugar, buscamos discretizar la energía de la ecuación 2.18:

$$TP_b^a(f) = \int_{S^1} \int_{S^1} \frac{\|T(u) \times (f(u) - f(v))\|^a}{\|f(u) - f(v)\|^b} du dv. \quad (3.28)$$

Sea $K_b^a(\mathbf{p}, \mathbf{q}, \mathbf{T}) = \|T \times (p - q)\|^a / \|p - q\|^b$, entonces la energía se expresa

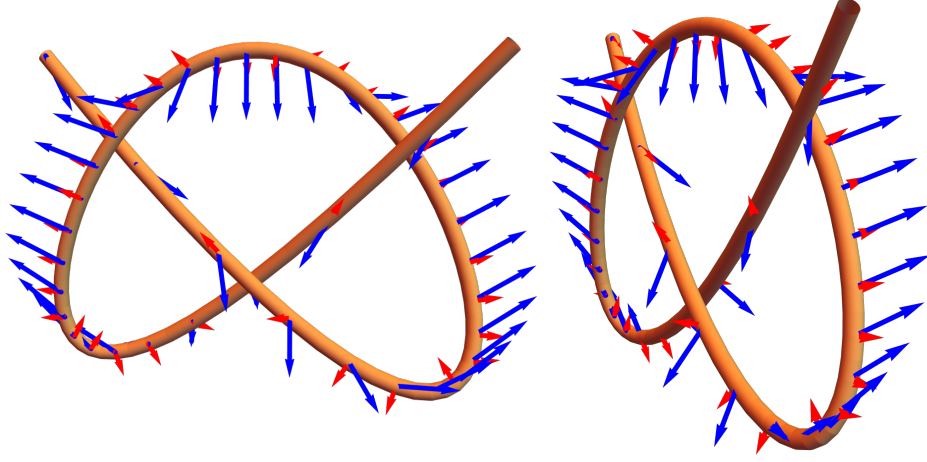


Figura 3.9: Comparación entre el flujo descrito por el gradiente convencional (rojo) y el gradiente fraccionario para la energía de la ecuación 2.18 con $p = 2$, $q = 4.5$ (azul). La longitud de los vectores es proporcional a su magnitud.

como

$$TP_b^a(f) = \int_{S^1} \int_{S^1} K_b^a(f(u), f(v), T(u)) \, dudv. \quad (3.29)$$

Como la energía está definida para una curva suave, para discretizarla es necesario primero discretizar la curva f . Sin pérdida de generalidad, supongamos que tenemos un conjunto de valores $\{\mathbf{v}_i\}_{i=0}^n \subset \mathbb{R}^3$. Podemos ver a la discretización de la curva f como una gráfica simple con vértices $V = \{\mathbf{v}_i\}_{i=0}^n$ y aristas $E = \{(i, i-1)\}_{i=1}^n$. Sea $I = (i, i-1) \in E$, $l_I = \|\mathbf{v}_i - \mathbf{v}_{i-1}\|$ y $T_I = (\mathbf{v}_i - \mathbf{v}_{i-1})/l_i$. Así, el primer paso para discretizar la energía es separar la integral en los vértices de la curva discretizada:

$$TP_b^a(f) = \sum_{I \in E} \sum_{J \in E} \int_{\bar{I}} \int_{\bar{J}} K_b^a(f(u), f(v), T(u)) \, du \, dv, \quad (3.30)$$

con \bar{I} el intervalo orientado sobre la arista I . Removiendo los términos que contribuyen energía infinita, correspondientes a aristas I, J que tienen un vértice en común, obtenemos la expresión

$$TP_b^a(f) = \sum_{\substack{I, J \in E \\ I \cap J = \emptyset}} \int_{\bar{I}} \int_{\bar{J}} K_b^a(f(u), f(v), T(u)) \, du \, dv. \quad (3.31)$$

Aplicando ahora una regla de cuadratura sencilla sobre los extremos de los

intervalo de integración, encontramos

$$TP_b^a(\mathbf{v}_0, \dots, \mathbf{v}_n) = \sum_{\substack{I, J \in E \\ I \cap J = \emptyset}} \left(\frac{l_i l_j}{4} \sum_{\substack{c=-1,0 \\ d=-1,0}} K_b^a(\mathbf{v}_{i+c}, \mathbf{v}_{j+d}, T_i) \right). \quad (3.32)$$

La expresión 3.32 ya está evaluada en una curva discretizada y no continua. Podemos calcular su gradiente respecto a cada una de de las $3n + 3$ coordenadas de $\{\mathbf{v}_i\}_{i=0}^n$:

$$\nabla TP_b^a = \left(\frac{\partial TP_b^a}{\partial v_{0x}}, \frac{\partial TP_b^a}{\partial v_{0y}}, \frac{\partial TP_b^a}{\partial v_{0z}}, \dots, \frac{\partial TP_b^a}{\partial v_{nz}} \right). \quad (3.33)$$

El gradiente fraccionario $g \in \mathbb{R}^{3n+3}$ será la solución a un sistema de $3n + 3$ ecuaciones lineales, dado por

$$(A \otimes Id_3)g = \nabla TP_b^a, \quad (3.34)$$

con \otimes el producto de Kronecker sobre operadores lineales. La matriz $A \in \mathbb{R}^{n \times n}$ corresponde a la discretización del operador de derivada fraccionaria. Dicha matriz se puede descomponer como la suma de dos matrices

$$A = B + B_0, \quad (3.35)$$

con B el término de la derivada fraccionaria y B_0 un término de orden bajo que actúa como regularización para que el operador de derivada sea invertible de forma única. La matriz B se puede construir mediante los siguientes incrementos: sea W_{IJ} definido como

$$W_{IJ} = \frac{l_I}{l_J} 4 \sum_{\substack{c=-1,0 \\ d=-1,0}} \frac{1}{\|\mathbf{v}_{i+c} - \mathbf{v}_{j+d}\|^{2\sigma+1}}, \quad (3.36)$$

con $\sigma = ((b-1)/a) - 1$. Entonces para cada par $I, J \in E$ con $I \cap J = \emptyset$ y para todo $c = 0, 1, d = 0, 1$, incrementamos la matriz B como

$$\begin{aligned} B_{i+c, i+d} &+= (-1)^{c+d} \frac{W_{IJ}}{l_I^2}, & B_{i+c, j+d} &-= (-1)^{c+d} \frac{W_{IJ}(T_I \cdot T_J)}{l_I l_J}, \\ B_{j+c, j+d} &+= (-1)^{c+d} \frac{W_{IJ}}{l_J^2}, & B_{j+c, i+d} &-= (-1)^{c+d} \frac{W_{IJ}(T_I \cdot T_J)}{l_I l_J}. \end{aligned} \quad (3.37)$$

De forma análoga la matriz B_0 se construye a partir de un Q_{IJ}^0 , definido como

$$W_{IJ}^0 = \frac{l_I}{l_J} 4 \sum_{\substack{c=-1,0 \\ d=-1,0}} \frac{K_4^2(\mathbf{v}_i, \mathbf{v}_j, T_I)}{\|\mathbf{v}_{i+c} - \mathbf{v}_{j+d}\|^{2\sigma+1}}. \quad (3.38)$$

Utilizando incrementos, la matriz simplemente se incrementa como

$$\begin{aligned} B_{i+c,i+d}^0 &+= W_{IJ}^0, & B_{i+c,j+d}^0 &-= W_{IJ}^0, \\ B_{j+c,j+d}^0 &+= W_{IJ}^0, & B_{j+c,i+d}^0 &-= W_{IJ}^0. \end{aligned} \quad (3.39)$$

Proyección sobre restricciones

Es común que haya restricciones impuestas que le quiten grados de libertad a la cadena. Así, debemos implementar que el algoritmo proyecte el gradiente fraccionario sobre una configuración que satisfaga las restricciones. Sin pérdida de generalidad, sea $\Phi : \mathbb{R}^{3n+3} \rightarrow \mathbb{R}^k$ una función tal que $k < 3n-3$ y $\Phi(\mathbf{v}_0, \dots, \mathbf{v}_n) = 0$ es el conjunto. Sea $C = J_\Phi \in \mathbb{R}^{k \times (3n+3)}$ la matriz Jacobiana de Φ . Así, encontrar el gradiente fraccionario $g' \in \mathbb{R}^{3n+3}$ que cumple las restricciones es equivalente a resolver el sistema de $(3n+3) + k$ ecuaciones lineales dado por

$$\begin{pmatrix} (A \otimes Id_3) & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} g' \\ \lambda \end{pmatrix} = \begin{pmatrix} \nabla T P_b^a \\ 0 \end{pmatrix}, \quad (3.40)$$

con $\lambda \in \mathbb{R}^k$ un conjunto de multiplicadores de Lagrange. Al ya tener calculado g' , podemos mover la cadena con la regla $\mathbf{v}' = \mathbf{v} - \tau g'$. Para verificar que en efecto \mathbf{v}' sigue cumpliendo con las restricciones deseadas, podemos evaluar $\Phi(\mathbf{v}')$. Sin pérdida de generalidad, puede darse el caso donde $\|\Phi(\mathbf{v}')\| \neq 0$, por lo que podemos volver a proyectar sobre el conjunto de restricciones resolviendo ahora el sistema

$$\begin{pmatrix} (A \otimes Id_3) & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} g'' \\ -\mu \end{pmatrix} = \begin{pmatrix} 0 \\ \Phi(\mathbf{v}') \end{pmatrix}. \quad (3.41)$$

Nuevamente, $\mu \in \mathbb{R}^k$ corresponde a multiplicadores de Lagrange. Al obtener g'' , podemos actualizar $\mathbf{v}' = \mathbf{v}' + g''$. Este sistema se puede continuar resolviendo hasta que se cumpla la condición de que $\|\Phi(\mathbf{v}')\| < \epsilon$.

Así, podemos ahora presentar la minimización mediante descenso de gradiente fraccionario en el algoritmo 8. En términos computacionales, tanto el gradiente C como $\nabla T P_a^b$ pueden calcularse de forma exacta utilizando

Algoritmo 8 Descenso de gradiente fraccionario

```

1: function DESGRADFRAC( $\mathbf{v}^{(0)}, \Phi, \text{tol}, \tau, \epsilon$ )
2:   Sea  $\mathbf{v} = \mathbf{v}^{(0)}$ 
3:   while  $f(\mathbf{v}) > \text{tol}$  do
4:     Calcular  $\nabla TP_a^b(\mathbf{v})$ 
5:     Obtener  $g'$  al solucionar el sistema de ecuaciones 3.40
6:     Sea  $\mathbf{v}' = \mathbf{v} - \tau g'$ 
7:     while  $\|\Phi(\mathbf{v}')\| > \epsilon$  do
8:       Obtener  $g''$  al solucionar el sistema de ecuaciones 3.41
9:       Sea  $\mathbf{v}' = \mathbf{v}' + g''$ 
10:    end while
11:    Sea  $\mathbf{v} = \mathbf{v}'$ 
12:  end while
13:  return  $\mathbf{v}$ 
14: end function

```

diferenciación automática con números hiperduales [55]. Los diversos sistemas de ecuaciones lineales son demasiado inestables para utilizar un método exacto, por lo que se recurre a un algoritmo iterativo como MINRES [56, 57].

Ya que este método está diseñado en particular para desenredar curvas suaves minimizando la energía de punto tangente, su aplicación a otras funciones está limitada y no tiene mucha aplicación. Remitemos al lector interesado en estudiar el uso de gradientes fraccionarios en problemas más generales de optimización a [58, 22].

3.4. Otras energías posibles

A pesar de que en la sección 2.5 hicimos una breve discusión sobre los funcionales de energía usados con un nudo, la discusión solo expuso a la energía de punto tangente $TP_b^a(f)$ como la mejor posibilidad. Además de la energía de punto tangente expresada en la ecuación 3.32. Otras posibles funciones a minimizar son el negativo del cuadrado de la distancia entre los vértices extremos de la cadena, también llamado máximo alcance (*max span*):

$$-\text{MaxSpan}(\mathbf{v}_0, \dots, \mathbf{v}_n) = -\|\mathbf{v}_0 - \mathbf{v}_n\|^2. \quad (3.42)$$

Esta energía resulta bastante importante ya que es posible demostrar que dada una cadena poligonal con enlaces de longitud fija y ángulo interno variable, entonces el máximo alcance de la cadena se alcanza en una configuración plana [25]. Otro caso interesante consiste en minimizar la distancia entre los penúltimos vértices, es decir

$$-\text{SpecialSpan}(\mathbf{v}_0, \dots, \mathbf{v}_n) = -\|\mathbf{v}_1 - \mathbf{v}_{n-1}\|^2. \quad (3.43)$$

Por otro lado, una energía que fue existosa para el caso de llevar cadenas 2D a configuraciones convexas es la energía de distancia elíptica (*elliptic-distance energy*) [59], expresada como

$$\text{Elliptic}(\mathbf{v}_0, \dots, \mathbf{v}_n) = \sum_{\substack{i,j \in E \\ k \in V \\ k \neq i,j}} \frac{1}{(\|\mathbf{v}_i - \mathbf{v}_k\| + \|\mathbf{v}_j - \mathbf{v}_k\| - \|\mathbf{v}_i - \mathbf{v}_j\|)}. \quad (3.44)$$

Notemos que cada término de la suma se vuelve constante cuando el vértice \mathbf{v}_k se desplaza en una elipse con focos $\mathbf{v}_i, \mathbf{v}_j$. Por otro lado, el término tiende a infinito cuando \mathbf{v}_k está contenido en el enlace (i, j) .

Por último, una energía que podría ser muy intuitiva sería una “distancia” o algún tipo de métrica entre la configuración actual y la configuración a la cual nos queremos acercar. Para construir dicha métrica, primero necesitamos poder quitar los grados de libertad debidos a movimientos rígidos de la cadena para de forma efectiva poder comparar dos cadenas.

Para quitarnos de esos grados de libertad, podemos comparar las cadenas desde sus coordenadas internas o centrar las cadenas (i.e. restarle el centroide a cada vértice), para quitarnos los grados de libertad rotacional, y luego aplicar el algoritmo de Kabsch [60] para encontrar la rotación óptima entre ambas cadenas. Así, ya teniendo las cadenas en sus configuraciones más coincidentes, podemos medir su diferencia mediante el RMSD (*Root mean square deviation*), dado por la expresión

$$\text{RMSD}(\mathbf{v}_0, \dots, \mathbf{v}_n, \mathbf{u}_0, \dots, \mathbf{u}_n) = \sqrt{\frac{1}{n+1} \sum_{i=0}^n \|\mathbf{v}_i - \mathbf{u}_i\|^2}. \quad (3.45)$$

Si pretendemos cambiar los ángulos internos y diedros, nuestra cadena de referencia puede ser la que contiene a todos los puntos en una recta, mientras que si solo movemos los diedros podemos tomar la configuración plana con los mismos ángulos internos actuales.

Capítulo 4

Resultados

Para poner a prueba el enfoque de minimización, intentaremos llevar dos cadenas poligonales, sobre las cuales conocemos resultados analíticos sobre su espacio de configuraciones, a configuraciones canónicas que deben serles accesibles. Estos dos ejemplos, presentados ya en la sección 2, son mostrados en la figura 4.1.

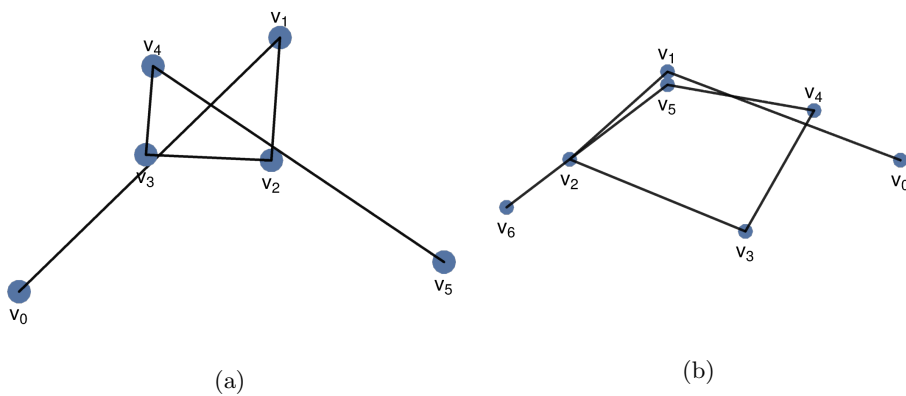


Figura 4.1: Cadenas que buscamos llevar a otra configuración con el proceso de minimización.

Para mayor simplicidad, nos referiremos a la cadena 4.1(a) como la cadena “**agujas**”, mientras que la cadena 4.1(b) será la cadena “**rectangular**”. La cadena de agujas tendrá como grados de libertad modificables sus ángu-

los internos y diedros mientras que la cadena rectangular solo podrá cambiar en sus ángulos diedros. Los detalles sobre cómo construir dichas cadenas y otras curvas que contienen nudos se pueden encontrar en el **anexo B a este trabajo**.

A continuación detallamos los hiperparámetros que se seleccionaron para los algoritmos de optimización y explicaremos como se mutó una cadena para genera nuevas configuraciones a partir de ella.

4.1. Mutaciones e hiperparámetros

En la sección anterior revisamos tres algoritmos que utilizaremos para intentar llevar las cadenas a una configuración estirada modificando solo algunos ángulos. En el caso del algoritmo de descenso de gradiente fraccionario, el movimiento está dado ya de forma natural. Sin embargo, para los algoritmos metaheurísticos (recocido simulado y algoritmos genéticos), es necesario especificar cómo haremos la mutación de la cadena.

Para poder permitir distintos grados de mutación sobre la cadena, el procedimiento de mutación será de la siguiente manera:

1. Dado un entero m , generamos un número aleatorio m_k entero escogido de forma uniforme entre 1, y m . m_k representa el número de grados de libertad que vamos a modificar de la cadena.
2. Posteriormente, generamos m_k parejas (j_i, α_i) con i un índice que represente algún ángulo interno o diedro de la cadena y α_i un ángulo tomado de forma aleatoria. Normalmente, $\alpha_i \sim N(0, \pi/8)$ y truncamos la distribución para garantizar que $\alpha \in [-\pi/40, \pi/40]$ ¹.

Así, la mutación está dada por cambiar primero el ángulo j_1 un valor α_1 , luego el ángulo j_2 un valor α_2 y así hasta llegar al ángulo j_{m_k} y el valor α_{m_k} . Así, aunque las mutaciones cambien varios grados de libertad de la cadena, en la práctica siempre cambiamos uno a la vez. Esto se debe de realizar así ya que verificar las intersecciones al mover más de un ángulo se torna demasiado complicado.

Geoméricamente, esto significa que las mutaciones solo cambian la configuración desplazándose sobre los ejes cartesianos dentro del espacio de

¹El valor de $\pi/40$ se escogió de forma heurística después de realizar algunas pruebas iniciales sobre el sistema.

configuraciones de la cadena. Esto podría parecer un problema ya que en principio nos gustaría poder hacer cambios arbitrarios. Sin embargo, para valores suficientemente pequeños de α_i , es posible aproximar cualquier movimiento sobre el espacio de configuraciones a través de movimientos sobre los ejes coordenados. La figura 4.2 ejemplifica esto.

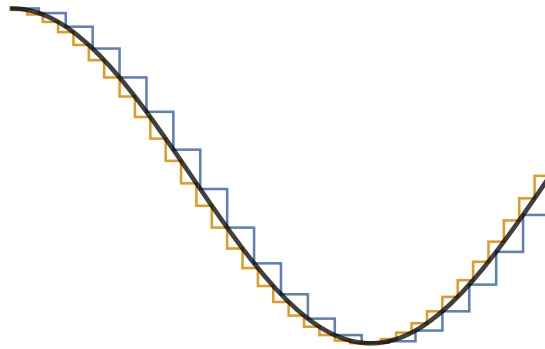


Figura 4.2: Aproximación de una curva a través de movimientos coordenados. La aproximación naranja tiene menor movimiento y es mejor a la curva.

Los hiperparámetros de cada simulación fueron los mismos para cada algoritmo. Los detallaremos a continuación

- **Genético:** selección de truncamiento con $k = 20$ y una población de 100, máximo 5000 iteraciones y mutaciones de hasta $m = 3$ grados de libertad.
- **Annealing:** temperatura inicial de 4.0 y final de 0.01 en unidades de la energía inicial de la cadena, programa exponencial de temperatura con $r = 0.99$, máximo 100000 o 500000 iteraciones y mutaciones de hasta $m = 3$ grados de libertad.
- **FGD:** paso de tiempo de 0.01 para curvas suaves y 0.001 para cadenas con menos de 10 puntos. 10 iteraciones máximas para la proyección sobre las restricciones, tolerancia de 10^{-4} de diferencia sobre las restricciones.

Dichos parámetros fueron seleccionados después de un ajuste manual. En principio, pueden ser refinados de diversas maneras.

El código para realizar las simulaciones fue implementado en el lenguaje Julia [61] y se encuentra disponible en GitHub en el repositorio <https://github.com/sayeg84/chainUnlocking.git>.

4.2. Dicción general

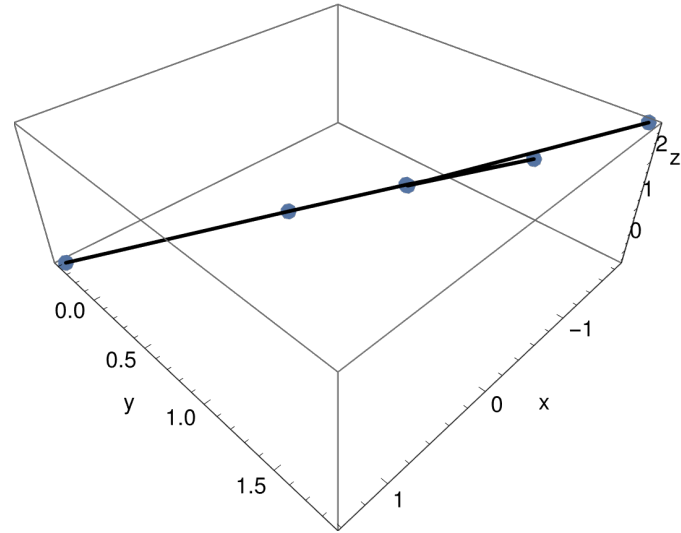
Las figuras B.1-B.4 del anexo B muestran el valor de la energía como función del paso de mutación para distintas cadenas. Sin embargo, la forma óptima de visualizar el proceso de desenredamiento mediante minimización de energía, más allá de los valores en sí que va generando la cadena, es mediante una animación que nos permita ver de forma visual si la cadena llega a la configuración deseada o se queda atrapada en otra región del espacio de configuraciones.

El lector puede consultar en el anexo B las tablas B.1 y B.2 que relacionan los videos de simulaciones para distintas cadenas, mediante distintos algoritmos y energías. En las figuras B.5-B.13 también incluimos algunos cuadros de simulaciones que nos parecieron especialmente relevantes. Al ver tanto las figuras B.1-B.4 como las animaciones del proceso, es posible concluir de inmediato algunas observaciones importantes.

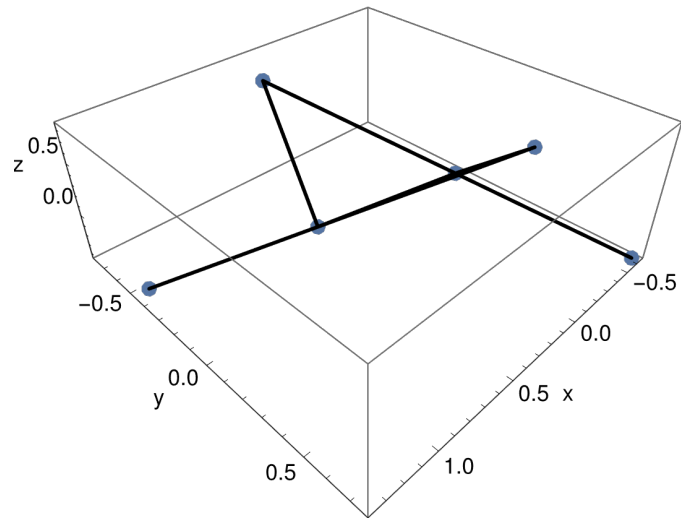
- La energía que presenta los mejores resultados es el MaxSpan y SpecialSpan para la cadena agujas y rectangular, respectivamente. Todas las demás energías no son óptimas ya que generalmente siempre se acerca a configuraciones que mantienen el nudo pero logran poner a una distancia grande los extremos. La energía de punto tangente es la única que logra resultados que se pueden aproximar los del MaxSpan y SpecialSpan.
- El procedimiento de desenredamiento es generalmente irrealizable mediante los algoritmos metaheurísticos más allá de casos sencillos. El único caso donde se observa claramente el desenredamiento de una cadena es para una aguja al minimizar la energía MaxSpan mediante el algoritmo de Annealing. Aunque los algoritmos genéticos parecen acercar las cadenas a dicho desenredamiento, se vuelve demasiado complicado que un paso de mutación aproxime bien el movimiento necesario para desenredarla, incluso con un número grande de iteraciones (más de 10000).
- El algoritmo de descenso de gradiente graccionario (FGD) no es útil

de realizar sobre cadenas con pocos vértices pues, aunque tomemos un paso de tiempo pequeño, es prácticamente imposible evitar las intersecciones debido a que no hay vértices suficientes para repeler dicha intersección.

- Es posible desenredar nudos que surgen sobre curvas suaves utilizando el algoritmo FGD. Es necesario que las curvas tengan suficientes vértices en las regiones enredadas para que la energía de punto tangente las pueda evitar las intersecciones. El paso de aprendizaje τ tiene una importancia pequeña en comparación con el número de vértices necesarios.



(a)



(b)

Figura 4.3: Últimas configuraciones para simulaciones con algoritmo genético sobre la cadena (a) agujas con $l = 2$ y (b) rectangular con $l = 1.15$.

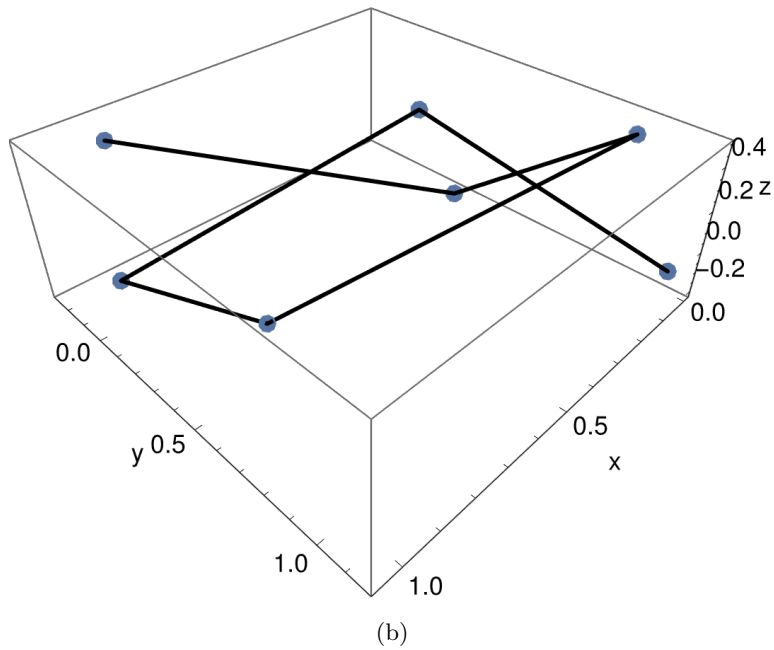
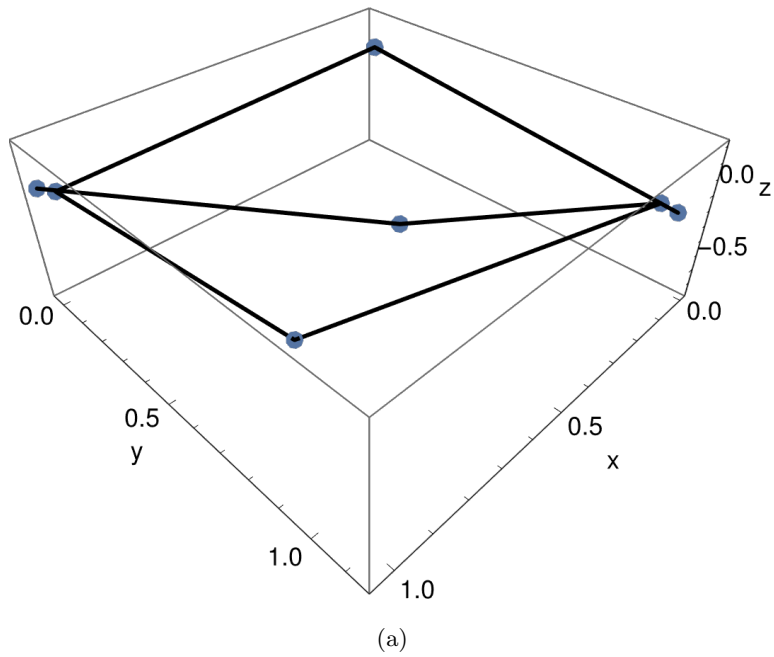


Figura 4.4: Últimas configuraciones para simulaciones con algoritmo genético sobre la cadena (a) agujas con $l = 2$ y (b) rectangular con $l = 1.15$.

Así, pareciera que el algoritmo que tiene éxito para desenredar nudos (FGD) no logra desenredar las cadenas por su bajo número de vértices. Una alternativa a dicho problema sería construir una versión de “alta resolución” de una cadena, en la cual pongamos muchos vértices sobre sus enlaces. La figura 4.5 muestra como sería la cadena de la figura 4.1(a) con vértices añadidos.

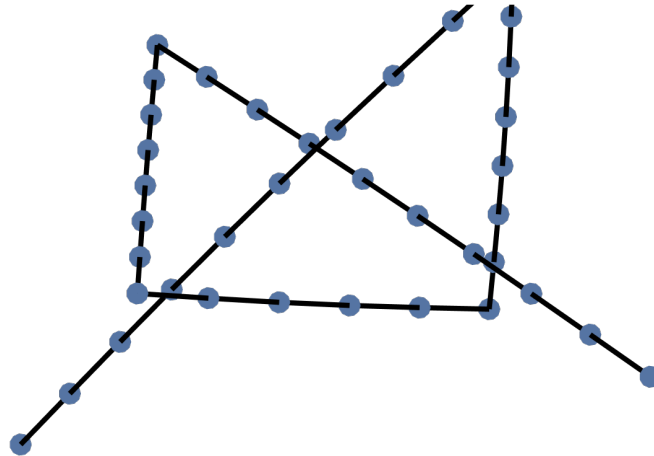


Figura 4.5: Cadena agujas en alta resolución.

Para mantener la estructura de la cadena original, los nuevos enlaces que se generan dentro de cada enlace original deben tener la restricción de no poder cambiar su longitud, sus ángulos internos o dihedros. Realizar la simulación mediante FGD de una cadena en esta alta resolución vuelve una simulación mucho más demandante, pues sabemos que resolver los sistemas de ecuaciones 3.40 y 3.41 tiene una complejidad cuadrática sobre el número de vértices. Notemos que la matriz jacobiana c también aumenta considerablemente en tamaño a la hora de hacer dicha simulación, lo que también vuelve más lenta la simulación.

Desafortunadamente, a la hora de implementar la simulación en alta resolución para la cadena “aguja” notamos que nos enfrentamos ahora al problema de que si bien la intersección se evita, las restricciones son demasiadas para permitir que la cadena se mueva en la forma deseada para desenredarse. Dicho problema puede visualizarse en el video de la tabla B.1.

4.3. Transición de encierro

Independientemente de que el desenredamiento solo se puede realizar para casos muy específicos, nos interesa analizar si es posible encontrar el momento a partir del cual se da la transición entre una cadena libre y una cadena encerrada. Para buscar dicho cambio, realizamos varias simulaciones de ambas cadenas de interés variando su longitud l entre sus enlaces extremos². Las simulaciones se realizaron utilizando solo los algoritmos metaheurísticos, con los mismos hiperparámetros que fueron detallados en la sección anterior, con la única diferencia de que para el annealing se tomaron 20 simulaciones independientes para cada valor de l .

Aunque en principio se debería de notar una transición en el valor promedio final de la energía como función de l cuando las cadenas puedan llegar a la configuración canónica, esta transición será más clara en el MaxSpan de la cadena. Así, además de analizar la energía, también revisamos al MaxSpan pues es un indicador de transición entre encerrado y no encerrado. En particular, ya que los algoritmos metaheurísticos llevan acabo varias iteraciones independientes de posibles soluciones, no debemos fijarnos solamente en el valor promedio del MaxSpan entre dichas soluciones sino en el valor máximo, pues es posible que algunas soluciones no lleguen a la configuración canónica pero basta con que una sola llegue para indicar que la cadena no está encerrada.

Para las simulaciones con la energía de punto tangente TP , se obtuvieron los siguientes resultados mostrados en las figuras 4.6 y 4.7.

² l es la longitud entre \mathbf{v}_0 y \mathbf{v}_1 y entre \mathbf{v}_{n-1} y \mathbf{v}_n

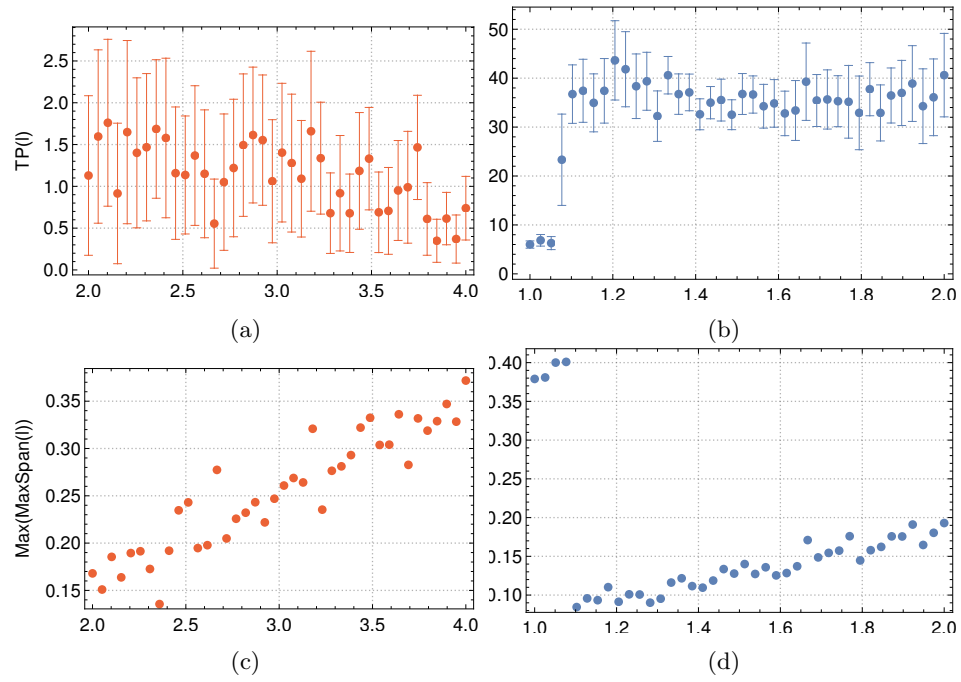


Figura 4.6: Resultados de simulación como función de l para la energía de punto tangente con el algoritmo Annealing. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.

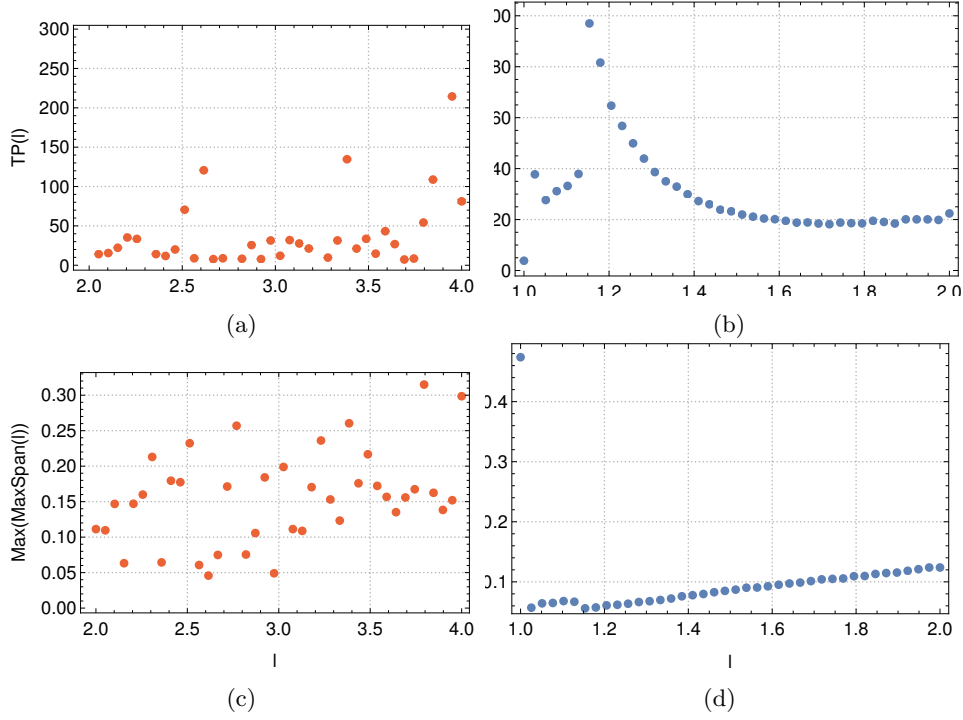


Figura 4.7: Resultados de simulación como función de l para la energía de punto tangente con el algoritmo genético. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.

En las subfiguras b) y d) de las figuras 4.6 y 4.7 podemos notar que se da una transición en los valores de la energía y del MaxSpan. Sabemos que los valores analíticos en los cuales supondríamos que es posible dar una diferencia es en $l_0 = 3$ para la cadena agujas y $l_0 = \sqrt{2}$ para la cadena rectangular. Ya que se ve de forma más clara para el algoritmo de annealing, podemos realizar otra simulación de forma refinada sobre el intervalo $l \in [1, 1.2]$ y realizar un ajuste con una curva sigmoial de la forma:

$$f(l) = \frac{a}{1 + \exp(-k(l - l_0))} + b, \quad (4.1)$$

con $a, b, k, l_0 \in \mathbb{R}$ los parámetros de la curva sigmoial. l_0 representa el valor donde se da la transición. Dichos ajustes se muestran en la figura 4.8

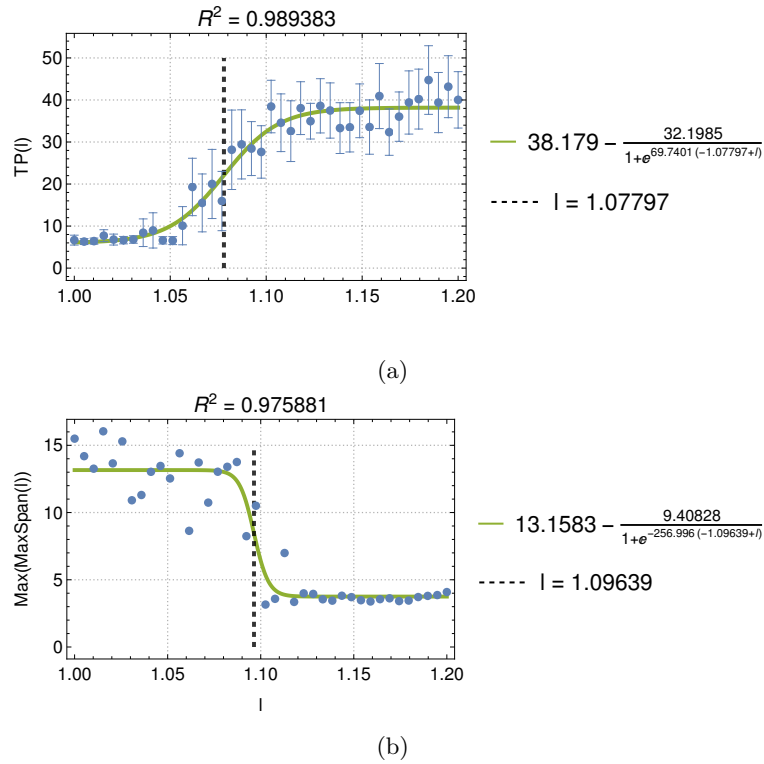


Figura 4.8: (a) Energía tangente promedio y (b) máximo del MaxSpan como función de l para la cadena rectangular con algoritmo Annealing.

De ambos ajustes obtenemos valores aproximados para la transición de $l_0 \approx 1.07797, 1.09639$, por lo que no es consistente con el valor analítico especulado. Cabe mencionarse que, al no existir una prueba formal sobre el valor analítico de l_0 para la cadena, es posible que la transición realmente se dé en este punto. Más aún, la transición es totalmente clara tanto al analizar la energía de punto tangente como el MaxSpan.

Al realizar la simulación utilizando como energía el MaxSpan para la cadena agujas y el SpecialSpan para la cadena rectangular, se obtienen los resultados mostrados en las figuras y

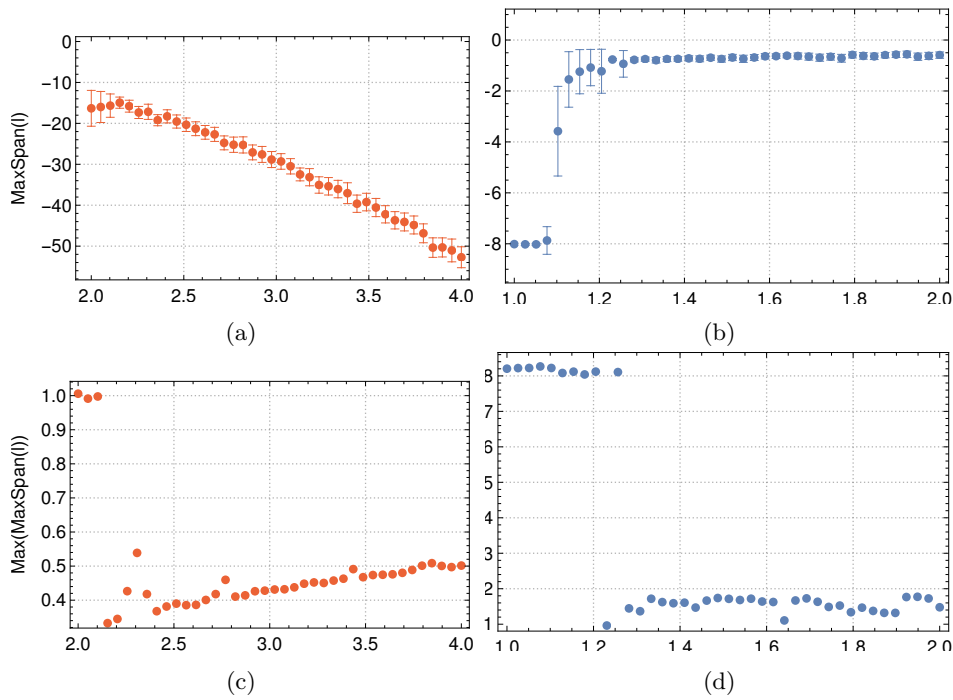


Figura 4.9: Resultados de simulación como función de l para la energía de MaxSpan con el algoritmo Annealing. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.

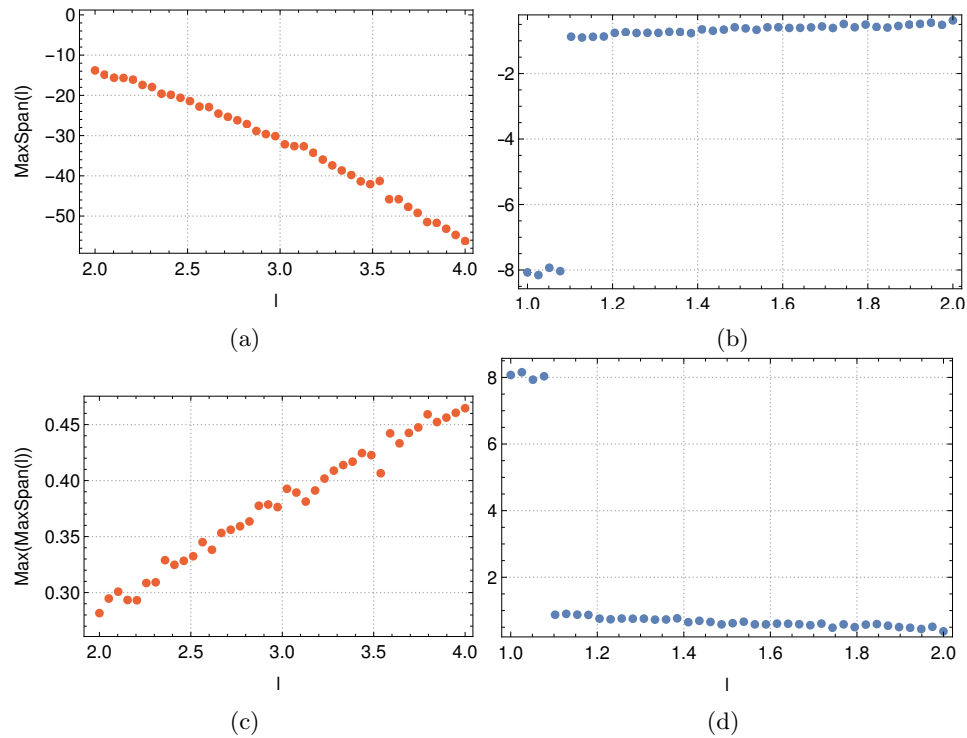


Figura 4.10: Resultados de simulación como función de l para la energía de MaxSpan con el algoritmo genético. (a,c) cadena agujas, (b,d) cadena rectangular. La energía y el MaxSpan están promediadas sobre las últimas 0.05 configuraciones.

Nuevamente, observamos la transición en las subfiguras b) y d) de las figuras 4.9 y 4.10. Más aún, ahora podemos observar una pequeña transición también en la subfigura c) de la figura 4.9. Nuevamente, para poder observar estas transiciones con mayor claridad, realizamos otras simulaciones sobre los intervalos de interés y un ajuste sigmoïdal. Para la cadena rectangular, realizamos dos refinamientos: uno para $l \in [1, 1.2]$, mostrado en la figura y otro para $l \in [1, 1.4]$, mostrado en la figura . En ambas simulaciones aumentamos el valor de la temperatura inicial a 15 (en unidades de la energía inicial) y tomamos un máximo de 300,000 pasos de mutación.

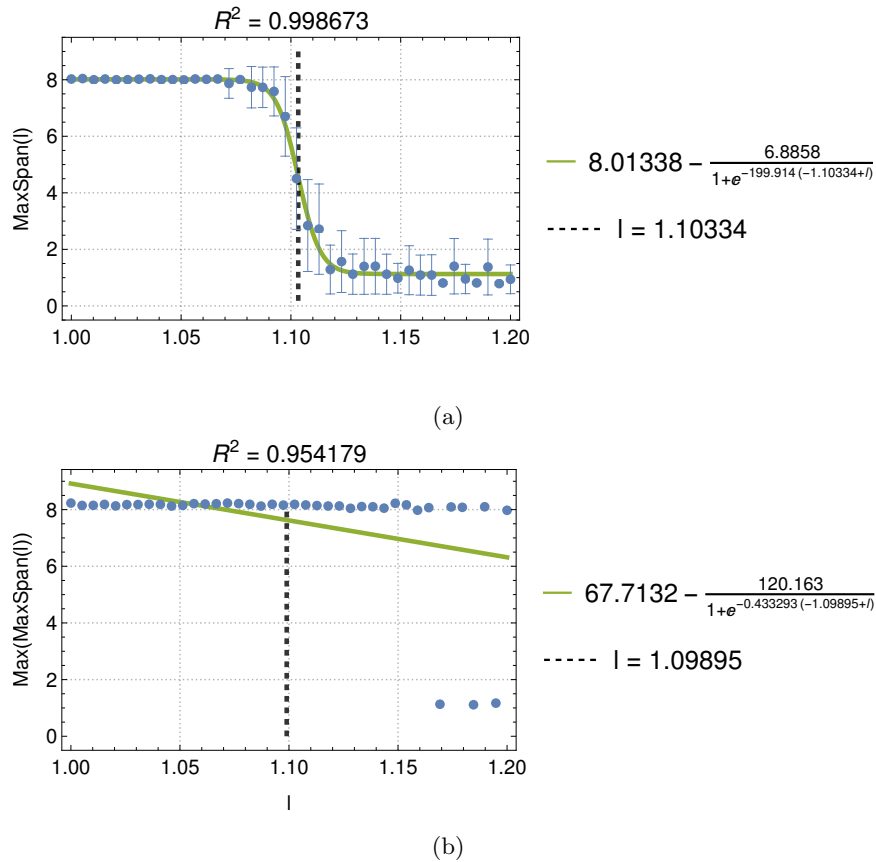
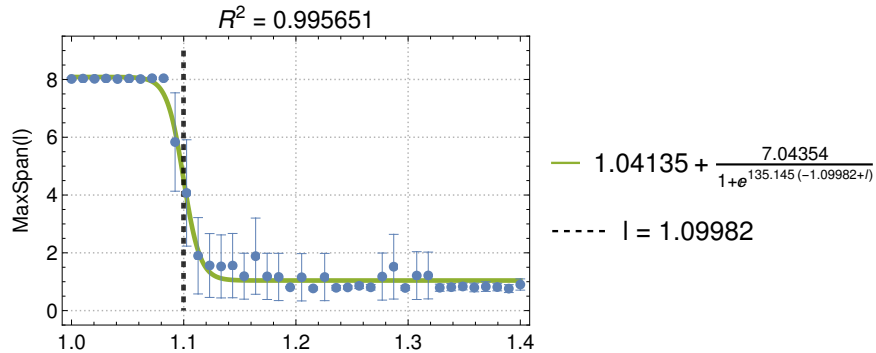
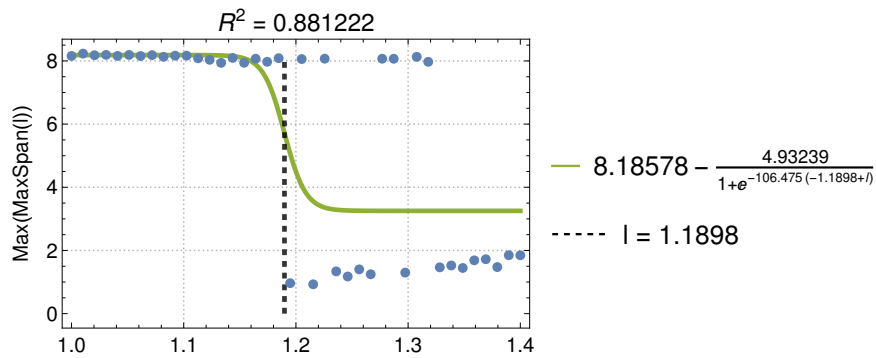


Figura 4.11: (a) SpecialSpan promedio y (b) máximo del SpecialSpan como función de l para la cadena rectangular con algoritmo Annealing con $l \in [1, 1.2]$

Aunque el ajuste para el valor máximo no es muy bueno, el ajuste del promedio es excelente. Obtenemos valores de transición en $l_0 = 1.10334, 1.09895$. Ya que la subgráfica b) de la figura 4.11 parece indicar que todavía no se da la transición, podemos analizar la figura 4.12 para un intervalo más grande de valores de l



(a)



(b)

Figura 4.12: (a) SpecialSpan promedio y (b) máximo del SpecialSpan como función de l para la cadena rectangular con algoritmo Annealing con $l \in [1, 1.4]$.

Los resultados de la figura 4.12 son muy parecidos a los de la figura 4.11. Nuevamente la transición no es clara para el máximo. Sin embargo, es claro que todavía hay valores que se logran desenredar para $l \approx 1.31$. El video de dicha cadena se puede consultar en la tabla B.1, así como en la figura B.11.

Para la cadena agujas, realizamos el mismo tipo de refinamiento con Annealing con temperatura inicial 15 (en unidades de la energía inicial) y 300,000 pasos máximos de mutación. Los resultados se muestra en la figura . A partir de la gráfica para el Máximo del MaxSpan, podemos notar que la cadena se desenreda todavía para valores de l cercanos a 2.6. Sin embargo, el ajuste no nos permite determinar con exactitud en cuál de estos valores

podría darse una transición.

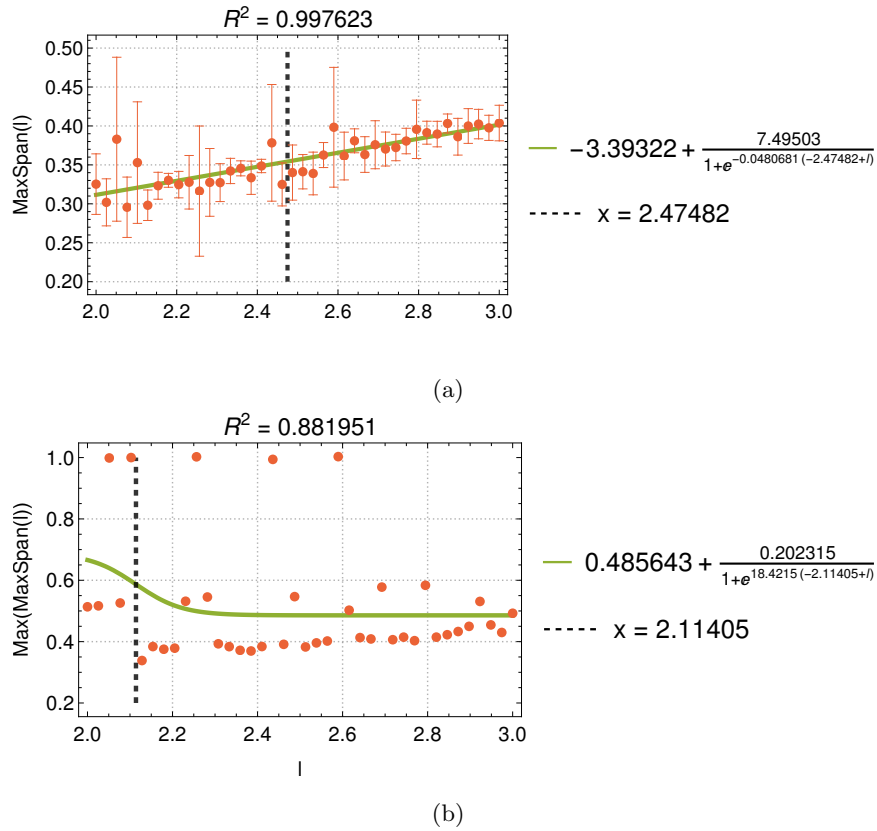


Figura 4.13: (a) MaxSpan promedio y (b) máximo del MaxSpan como función de l para la cadena agujas con algoritmo Annealing.

Capítulo 5

Conclusiones

La teoría sobre las cadenas poligonales y su uso para modelar la cadena principal de una proteína es una teoría sólida. Aunque el enfoque de minimización de energía se puede usar con éxito en teoría de nudos y, en particular, el algoritmo FGD muestra muy impresionantes resultados para desdoblar estructuras curvas, el enfoque no se puede utilizar de forma tan sencilla para desdoblar cadenas poligonales.

Los algoritmos metaheurísticos no pueden lograr llevar las cadenas a su configuración canónica incluso en casos sencillos. Esto se puede explicar debido a que probablemente la aproximación mediante movimientos sobre los ejes coordenados en el espacio de configuraciones no es suficientemente buena como para realmente aproximar la trayectoria por la que es necesario llevar una cadena hasta otra configuración.

Interpretando de forma probabilística, podríamos argumentar que en realidad un algoritmo metaheurístico tiene más bien una probabilidad de llevar la cadena a dicha configuración, y que esta probabilidad cambia de forma proporcional a la parametrización de la cadena. Así, un algoritmo con un número infinito de iteraciones y otros hiperparámetros (como una población o iteraciones independientes suficientemente numerosas, una temperatura inicial infinita para el Annealing, cambios infinitesimalmente pequeños en los grados de libertad) siempre desdoblaría la cadena cuando las longitudes de los enlaces lo permitieran. Esto es consistente con las condiciones de convergencia que tienen varios algoritmos metaheurísticos.

Es posible llevar más lejos este argumento y decir que, si bien hay muchos factores físicos y químicos que afectan la forma en la que una proteína se mueve por su espacio de configuraciones, es posible que haya restricciones geométricas que limitan drásticamente la probabilidad de encontrar a una proteína en cierta región del espacio de configuraciones. Dicha restricción podría explicar parcialmente la discrepancia entre los tiempos de plegamiento y la cantidad de configuraciones accesibles.

Algunas mejoras que se pueden realizar sobre este trabajo son:

1. Encontrar la forma de realizar la reproducción de dos cadenas para que el algoritmo genético pueda explorar de forma más profunda el espacio de configuraciones de la cadena poligonal. Esto podría haciendo promediando las coordenadas internas entre ambas cadenas. El hecho de que ambas partan de la misma componente conexa en el espacio de configuraciones podría garantizar que se van a mantener ahí.
2. Utilizar funciones de energía que no sean genéricas sino específicas para cada tipo de cadena. Por ejemplo, para la cadena rectangular, podríamos penalizar la cercanía de los vértices \mathbf{v}_1 y \mathbf{v}_5 , así como penalizar la cercanía entre \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 y \mathbf{v}_4 para el caso de las agujas. La problemática de realizar esto es perder la generalidad del método.
3. Modificar la implementación del algoritmo FGD para intentar realizarlo directamente sobre el espacio de configuraciones de la cadena y no sobre la posición geométrica de los vértices. Más aún, es posible cambiar la forma de hacer la simulación en alta resolución para acelerar el tiempo de solución de las ecuaciones lineales.
4. Utilizar un algoritmo de geometría computacional para revisar si se da la intersección al rotar un segmento. Una propuesta de esto es proyectar todos los segmentos sobre un plano con dirección arbitraria y si las proyecciones $2D$ de los segmentos sobre dicho plano pueden rotarse sin intersectar, entonces es posible cambiar ese grado de libertad. Dicho problema se puede resolver con complejidad $n \log n$ sobre el número de segmentos.

Referencias

- [1] Thomas Little Heath y col. *The thirteen books of Euclid's Elements*. Cambridge University Press, 1968.
- [2] Mariano Nieto Viejobueno. *¿Existen atajos en geometría?* 2014. URL: <https://marianonieto.wordpress.com/2014/11/02/existen-atajos-en-geometria/comment-page-1/> (visitado 07-04-2022).
- [3] Andrew J. Hanson. *Calabi-Yau Space. Wolfram Demonstrations Project*. 2011. URL: <http://demonstrations.wolfram.com/CalabiYauSpace/> (visitado 07-04-2022).
- [4] Vincenzo De Risi. *Mathematizing Space: The objects of geometry from antiquity to the early modern age*. Birkhäuser, 2015.
- [5] Immanuel Kant. *Crítica de la razón pura*. RBA Gredos, 2015.
- [6] David Hilbert. *The foundations of geometry*. Open Court Publishing Company, 1950.
- [7] Albert Einstein. “Physical Meaning of Geometrical Propositions”. En: *Relativity: the special and general theory*. 3.^a ed. Methuen y Co., 1916.
- [8] Georges Gonthier y col. “Formal proof—the four-color theorem”. En: *Notices of the AMS* 55.11 (2008), págs. 1382-1393.
- [9] Thomas C Hales. “A proof of the Kepler conjecture”. En: *Annals of mathematics* 162.3 (2005), págs. 1065-1185.
- [10] Aleksandar Donev, Salvatore Torquato y Frank H Stillinger. “Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles. I. Algorithmic details”. En: *Journal of computational physics* 202.2 (2005), págs. 737-764.
- [11] Jonathan Shewchuk. “What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint)”. En: *University of California at Berkeley 2002* (2002).

- [12] *Structure of DNA (DeoxyriboNucleic Acid)*. 2011. URL: <https://commons.wikimedia.org/w/index.php?curid=15027555> (visitado 10-04-2022).
- [13] Cyrus Levinthal. “Are there pathways for protein folding?” En: *Journal de chimie physique* 65 (1968), págs. 44-45.
- [14] Erik D Demaine y Joseph O’Rourke. *Geometric folding algorithms: linkages, origami, polyhedra*. Cambridge University Press, 2007.
- [15] Joseph O’Rourke. *How to fold it: the mathematics of linkages, origami, and polyhedra*. Cambridge University Press, 2011.
- [16] Jun O’hara. *Energy of knots and conformal geometry*. Vol. 33. world scientific, 2003.
- [17] Simon Blatt y Philipp Reiter. “Modeling repulsive forces on fibres via knot energies”. En: *Computational and Mathematical Biophysics* 2.1 (2014), págs. 56-72.
- [18] Chris Yu, Henrik Schumacher y Keenan Crane. “Repulsive curves”. En: *ACM Transactions on Graphics (TOG)* 40.2 (2021), págs. 1-21.
- [19] B Suman y P Kumar. “A survey of simulated annealing as a tool for single and multiobjective optimization”. En: *Journal of the Operational Research Society* 57.10 (oct. de 2006), págs. 1143-1160. DOI: [10.1057/palgrave.jors.2602068](https://doi.org/10.1057/palgrave.jors.2602068). URL: <https://doi.org/10.1057/palgrave.jors.2602068>.
- [20] Terry J Ligocki y James A Sethian. “Recognizing knots using simulated annealing”. En: *Journal of Knot Theory and Its Ramifications* 3.04 (1994), págs. 477-495.
- [21] Sourabh Katoch, Sumit Singh Chauhan y Vijay Kumar. “A review on genetic algorithm: past, present, and future”. En: *Multimedia Tools and Applications* 80.5 (oct. de 2020), págs. 8091-8126. DOI: [10.1007/s11042-020-10139-6](https://doi.org/10.1007/s11042-020-10139-6). URL: <https://doi.org/10.1007/s11042-020-10139-6>.
- [22] Yiheng Wei y col. “Design of generalized fractional order gradient descent method”. En: *arXiv preprint arXiv:1901.05294* (2018).
- [23] Jason Cantarella y Heather Johnston. “Nontrivial Embeddings Of Polygonal Intervals and Unknots in 3-Space”. En: *Journal of Knot Theory and Its Ramifications* 07.08 (dic. de 1998), págs. 1027-1039. DOI: [10.1142/s0218216598000553](https://doi.org/10.1142/s0218216598000553). URL: <https://doi.org/10.1142/s0218216598000553>.

- [24] N. Benbernou. “Fixed-Angle Polygonal Chains: Locked Chains and the Maximum Span”. Bachelor’s Thesis. 2006. URL: <https://api.semanticscholar.org/CorpusID:122803766>.
- [25] Nadia Benbernou y Joseph O’Rourke. “On the maximum span of fixed-angle chains”. En: *arXiv preprint arXiv:0801.0258* (2008).
- [26] Michael R Garey y David S Johnson. *Computers and intractability*. 1.^a ed. Vol. 174. W.H. Freeman y Company, 1979.
- [27] C.H. Papadimitriou. *Computational Complexity*. Theoretical computer science. Addison-Wesley, 1994. ISBN: 9780201530827.
- [28] Michael Soss y Godfried T. Toussaint. “Geometric and computational aspects of polymer reconfiguration”. En: *Journal of Mathematical Chemistry* 27.4 (2000), págs. 303-318. DOI: [10.1023/a:1018823806289](https://doi.org/10.1023/a:1018823806289). URL: <https://doi.org/10.1023/a:1018823806289>.
- [29] David L Nelson, Albert L Lehninger y Michael M Cox. *Lehninger principles of biochemistry*. Macmillan, 2008.
- [30] Bengt Nölting. *Protein folding kinetics: biophysical methods*. Springer Science & Business Media, 2005.
- [31] Hans Frauenfelder. *The physics of proteins: an introduction to biological physics and molecular biophysics*. Springer Science & Business Media, 2010.
- [32] David E Mortenson y col. “Quasiracemic crystallization as a tool to assess the accommodation of noncanonical residues in natively protein conformations”. En: *Journal of the American Chemical Society* 134.5 (2012), págs. 2473-2476.
- [33] Erik D Demaine, Stefan Langerman y Joseph O’Rourke. “Geometric restrictions on producible polygonal protein chains”. En: *Algorithmica* 44.2 (2006), págs. 167-181.
- [34] *The structure and Function of the Ribosome*. 2016. URL: <https://www.flickr.com/photos/132318516@N08/24315920330> (visitado 28-02-2022).
- [35] Greg Aloupis y col. “Flat-state connectivity of linkages under dihedral motions”. En: *International Symposium on Algorithms and Computation*. Springer. 2002, págs. 369-380.

- [36] Allan Gut. “Random Variables”. En: *Probability: A Graduate Course: A Graduate Course*. New York, NY: Springer New York, 2013, págs. 25-117. DOI: [10.1007/978-1-4614-4708-5_2](https://doi.org/10.1007/978-1-4614-4708-5_2). URL: https://doi.org/10.1007/978-1-4614-4708-5_2.
- [37] *Projection from Product Topology is Open - Proof Wiki*. 2022. URL: https://www.proofwiki.org/wiki/Projection_from_Product_Topology_is_Open (visitado 14-02-2022).
- [38] James Munkres. *Topology*. 2.^a ed. Pearson, 2014.
- [39] WB Raymond Lickorish. *An introduction to knot theory*. Vol. 175. Springer Science & Business Media, 2012.
- [40] Kunio Murasugi. *Knot theory and its applications*. Springer Science & Business Media, 2007.
- [41] Marc Lackenby. *Elementary knot theory*. 2016. arXiv: [1604.03778](https://arxiv.org/abs/1604.03778) [[math.GT](https://arxiv.org/abs/1604.03778)].
- [42] Jun O’hara. “Energy of a knot”. En: *Topology* 30.2 (1991), págs. 241-247.
- [43] Michael A. Soss. “Geometric and Computational Aspects of Molecular Reconfiguration”. AAINQ75679. Tesis doct. CAN, 2001. ISBN: 0612756793.
- [44] Ivan J. Balaban. “An Optimal Algorithm for Finding Segments Intersections”. En: *Proceedings of the Eleventh Annual Symposium on Computational Geometry*. SCG ’95. Vancouver, British Columbia, Canada: Association for Computing Machinery, 1995, págs. 211-219. ISBN: 0897917243. DOI: [10.1145/220279.220302](https://doi.org/10.1145/220279.220302). URL: <https://doi.org/10.1145/220279.220302>.
- [45] Donald E Knuth. *The Art of Computer Programming, volume 2: Seminumerical Algorithms*. 3.^a ed. Addison-Wesley Professional, 1997.
- [46] *List of arbitrary-precision arithmetic software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=List_of_arbitrary-precision_arithmetic_software&oldid=1057518897. [Online; accessed 30-December-2021]. 2021.
- [47] Mykel J Kochenderfer y Tim A Wheeler. *Algorithms for optimization*. Mit Press, 2019.
- [48] *Numerical Optimization*. New York, NY: Springer New York, 2006. ISBN: 978-0-387-40065-5. DOI: [10.1007/978-0-387-40065-5](https://doi.org/10.1007/978-0-387-40065-5). URL: <https://doi.org/10.1007/978-0-387-40065-5>.

- [49] M.E.J. Newman y G.T. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1999. ISBN: 9780198517979.
- [50] Christian P. Robert y George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2005. ISBN: 0387212396.
- [51] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. Vol. 74. John Wiley & Sons, 2009.
- [52] M. Locatelli. “Simulated Annealing Algorithms for Continuous Global Optimization: Convergence Conditions”. En: *Journal of Optimization Theory and Applications* 104.1 (ene. de 2000), págs. 121-133. DOI: [10.1023/a:1004680806815](https://doi.org/10.1023/a:1004680806815). URL: <https://doi.org/10.1023/a:1004680806815>.
- [53] G. Rudolph. “Convergence of evolutionary algorithms in general search spaces”. En: *Proceedings of IEEE International Conference on Evolutionary Computation*. 1996, págs. 50-54. DOI: [10.1109/ICEC.1996.542332](https://doi.org/10.1109/ICEC.1996.542332).
- [54] Eleonora Di Nezza, Giampiero Palatucci y Enrico Valdinoci. “Hitchhiker’s guide to the fractional Sobolev spaces”. En: *Bulletin des Sciences Mathématiques* 136.5 (jul. de 2012), págs. 521-573. DOI: [10.1016/j.bulsci.2011.12.004](https://doi.org/10.1016/j.bulsci.2011.12.004). URL: <https://doi.org/10.1016/j.bulsci.2011.12.004>.
- [55] Jeffrey Fike y Juan Alonso. “The Development of Hyper-Dual Numbers for Exact Second-Derivative Calculations”. En: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. DOI: [10.2514/6.2011-886](https://doi.org/10.2514/6.2011-886). eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2011-886>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2011-886>.
- [56] C. C. Paige y M. A. Saunders. “Solution of Sparse Indefinite Systems of Linear Equations”. En: *SIAM Journal on Numerical Analysis* 12.4 (1975), págs. 617-629. DOI: [10.1137/0712047](https://doi.org/10.1137/0712047). eprint: <https://doi.org/10.1137/0712047>. URL: <https://doi.org/10.1137/0712047>.
- [57] *MINRES - IterativeSolvers.jl*. 2017. URL: https://iterativesolvers.julia.linearalgebra.org/stable/linear_systems/minres/#MINRES (visitado 07-04-2022).

- [58] Shujaat Khan y col. “A Novel Fractional Gradient-Based Learning Algorithm for Recurrent Neural Networks”. En: *Circuits, Systems, and Signal Processing* 37.2 (mayo de 2017), págs. 593-612. DOI: [10.1007/s00034-017-0572-z](https://doi.org/10.1007/s00034-017-0572-z). URL: <https://doi.org/10.1007/s00034-017-0572-z>.
- [59] Jason H Cantarella y col. “An energy-driven approach to linkage unfolding”. En: *Proceedings of the twentieth annual symposium on Computational geometry*. 2004, págs. 134-143.
- [60] W. Kabsch. “A discussion of the solution for the best rotation to relate two sets of vectors”. En: *Acta Crystallographica Section A* 34.5 (sep. de 1978), págs. 827-828. DOI: [10.1107/s0567739478001680](https://doi.org/10.1107/s0567739478001680). URL: <https://doi.org/10.1107/s0567739478001680>.
- [61] Jeff Bezanson y col. “Julia: A fresh approach to numerical computing”. En: *SIAM review* 59.1 (2017), págs. 65-98.
- [62] *Test functions for optimization* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 12-March-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Test_functions_for_optimization&oldid=1072363516.

Anexo A: Ejemplos de optimización

Las funciones a optimizar, con sus respectivos mínimos y dominios, se muestran en las siguientes ecuaciones [62].

$$1) \text{Eggholder}(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$

$$\text{Eggholder}(512, 404.2319) = -959.640$$

$$X = \{(x, y) \in \mathbb{R}^2 \mid -512 \leq x, y \leq 512\}.$$

$$2) \text{Mishra}(x, y) = \sin(y) \exp[(1 - \cos(x))^2] + \cos(x) \exp[(1 - \sin(y))^2] + (x - y)^2$$

$$\text{Mishra}(-3.1302468, -1.5821422) = -106.7645267$$

$$X = \{(x, y) \in \mathbb{R}^2 \mid (x + 5)^2 + (y + 5)^2 < 25\}.$$

$$3) \text{Gomez-levy}(x, y) = 4x^2 - 2.1x^4 + \frac{x^6}{3} + xy - 4y^2 = 4y^4$$

$$\text{Gomez-levy}(0.08984201, -0.7126564) = -2.0239884$$

$$X = \{(x, y) \in \mathbb{R}^2 \mid -\sin(4\pi x) + 2\sin^2(2\pi y) \leq 1.5, -1 \leq x, y \leq 1\}.$$

La regla para actualizar las posibles soluciones x fue $x' = x + Z$ con $Z \sim N(0, \eta)$ un vector aleatorio con distribución normal. Los hiperparámetros y otros ajustes de la optimización realizada se muestran en la tabla 5.1 para el annealing y en la tabla 5.2 para el algoritmo genético. Cabe mencionarse que, debido a que para una función el dominio no es convexo, se tomó solo

un padre y no se realizó reproducción en cada paso de selección. Ya que normalmente la reproducción en los algoritmos genéticos provoca que las soluciones convergan a un solo punto y no sigan explorando el dominio de forma aleatoria, el parámetro η se disminuyo exponencialmente para evitar que continuaran caminando.

Función	T_0	Regla de actualización	T_{min}	N_T	η
Eggholder	10^5	$T_{k+1} = 0.999T_k$	5	500	50
Mishra	100	$T_{k+1} = 0.99T_k$	10^{-4}	200	0.1
Gomez-levy	30	$T_{k+1} = 0.999T_k$	10^{-4}	300	0.2

Tabla 5.1: Tabla con hiperparámetros del recocido simulado para las funciones.

Función	Tamaño de población	Método de selección	Iteraciones máximas	η
Eggholder	200	Truncamiento, $k = 20$	2000	50
Mishra	100	Torneo, $k = 10$	200	0.1
Gomez-levy	100	Torneo, $k = 10$	200	0.1

Tabla 5.2: Tabla con hiperparámetros del recocido simulado para las funciones.

En las figuras [A.1](#), [A.4](#) y [A.7](#) mostramos primero las gráficas de las funciones (color azul indica valores más pequeños, color rojo valores más altos). Posteriormente, las figuras [A.2](#), [A.3](#), [A.5](#), [A.6](#), [A.8](#) y [A.9](#) algunos cuadros del proceso de minimización. Los puntos verdes representan soluciones independientes para el annealing y elementos de la población para el algoritmo genético. Para el annealing, el título de la figura describe la temperatura en ese cuadro, mientras que para el algoritmo genético el título describe la iteración del algoritmo. En la descripción de cada video hay un enlace para ver un video del proceso de minimización.

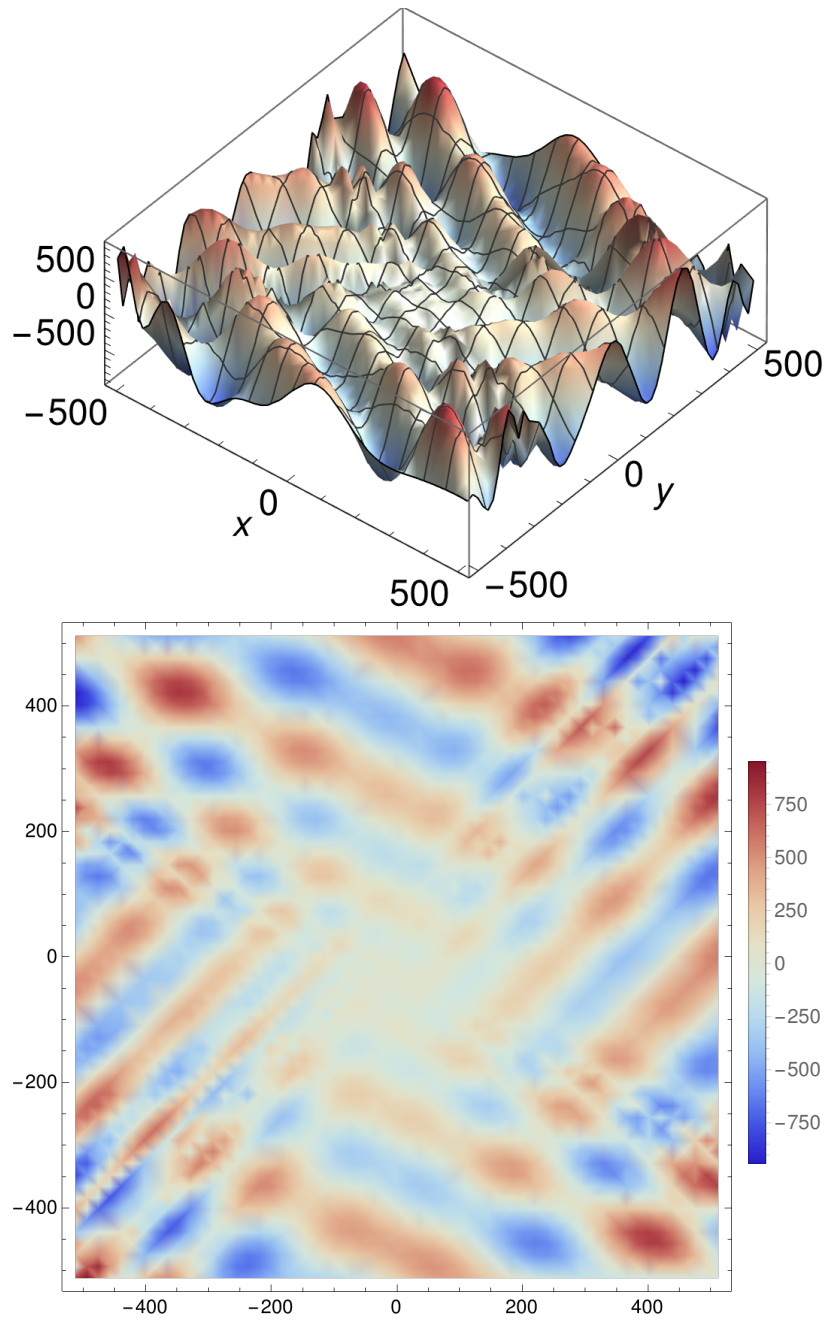


Figura A.1: Gráfica de la función Eggholder en \mathbb{R}^3 y como mapa de calor.

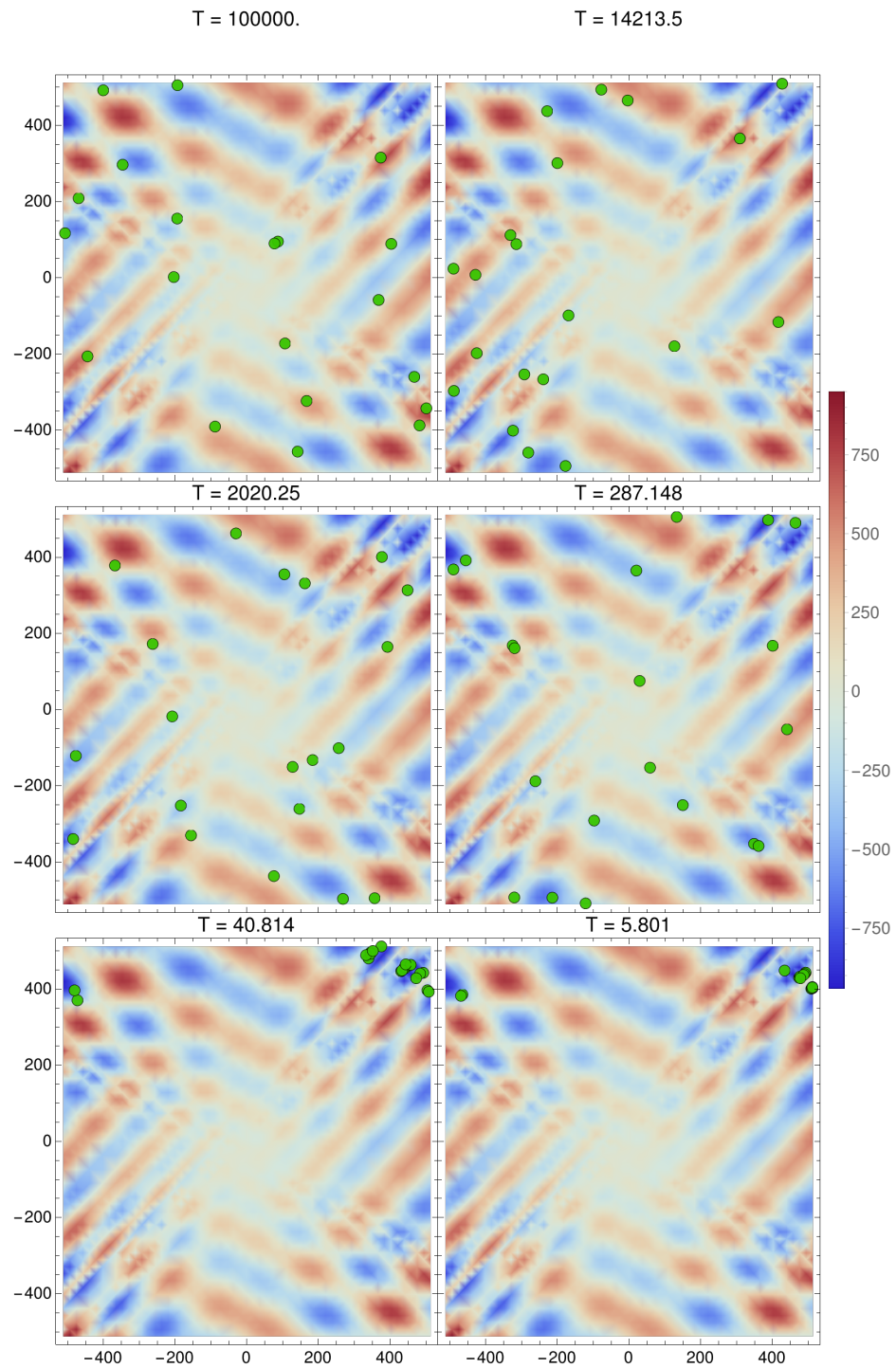


Figura A.2: Minimización de la función Eggholder mediante recocido simulado. Una animación del proceso puede encontrarse en <https://youtu.be/20TV63bm3Gk>.

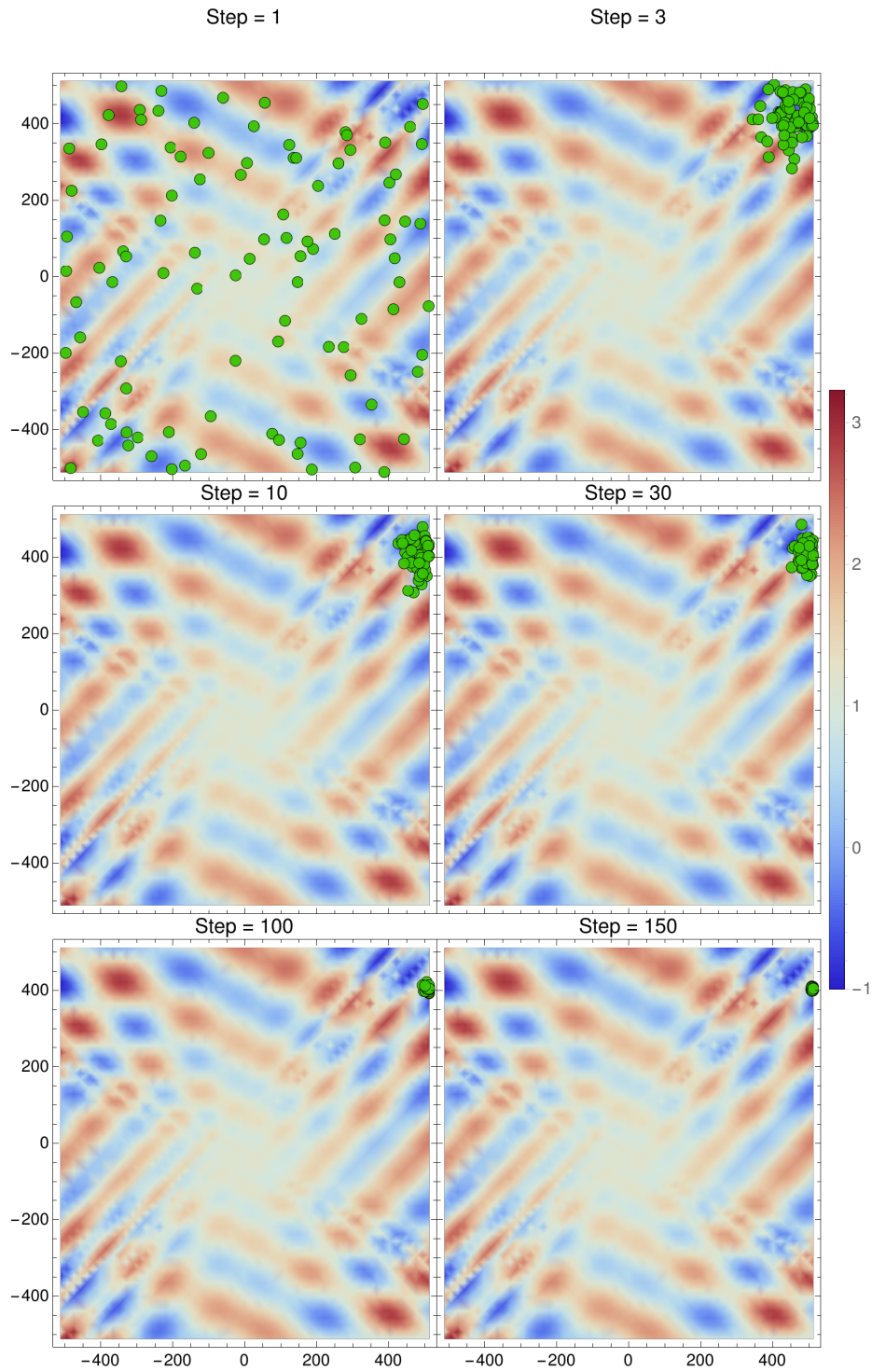
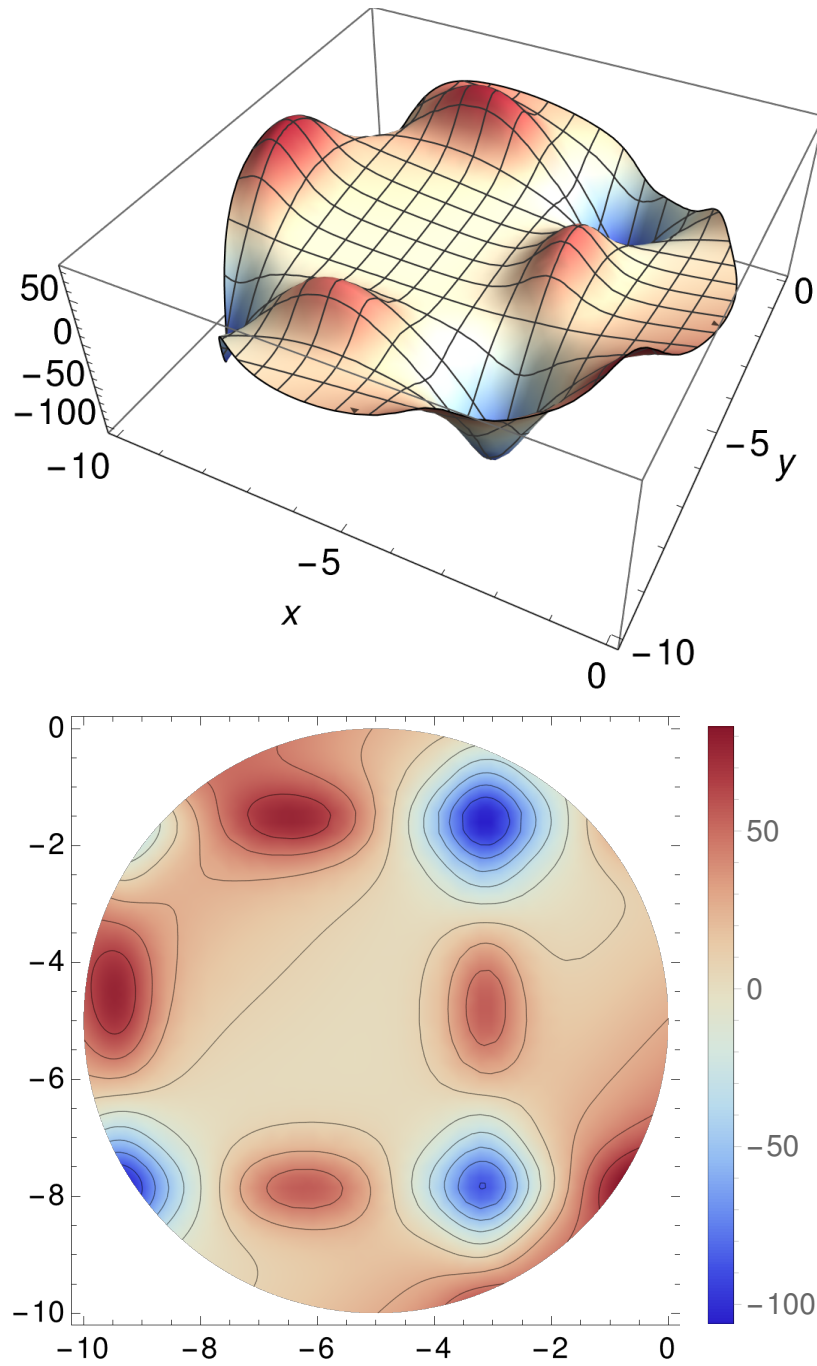


Figura A.3: Minimización de la función Eggholder mediante un algoritmo genético. Una animación del proceso puede encontrarse en https://youtu.be/OvmFI_bhOWI.

Figura A.4: Gráfica de la función de Mishra en \mathbb{R}^3 y como mapa de calor

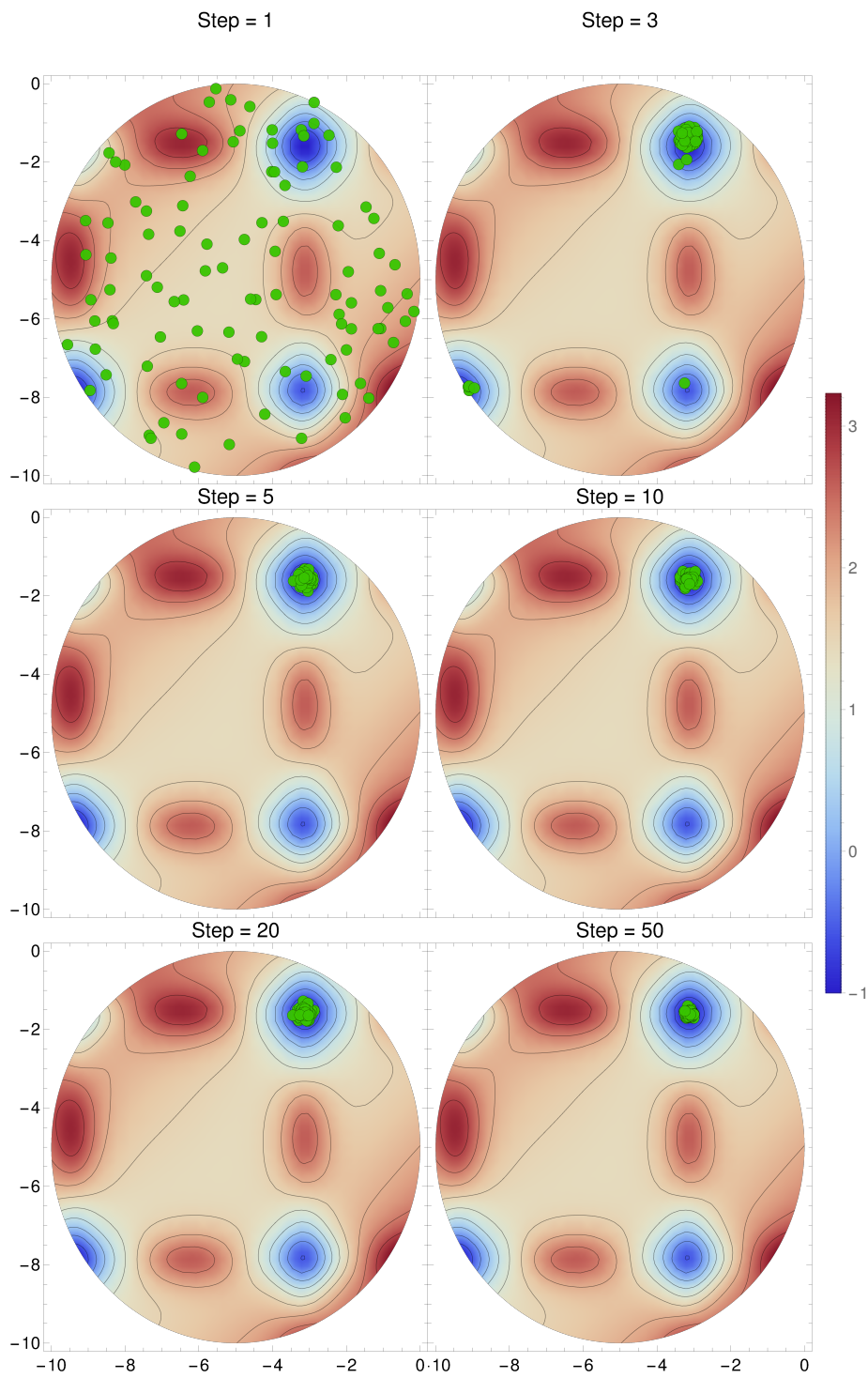


Figura A.5: Minimización de la función de Mishra mediante recocido simulado. Una animación del proceso puede encontrarse en https://youtu.be/3YB_0Br0wWQ.

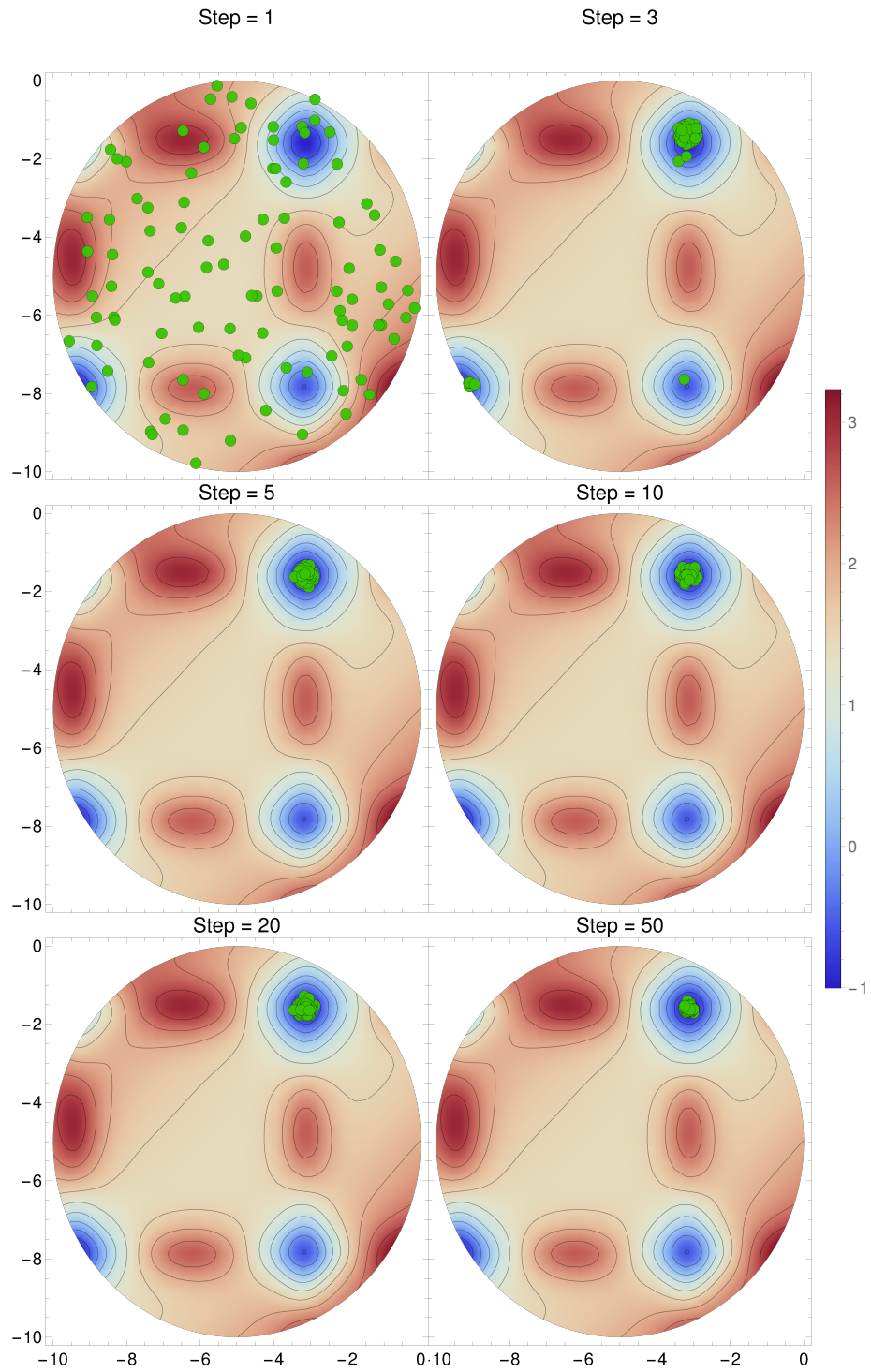


Figura A.6: Minimización de la función Mishra mediante un algoritmo genético. Una animación del proceso puede encontrarse en <https://youtu.be/s72096Y7nwQ>.

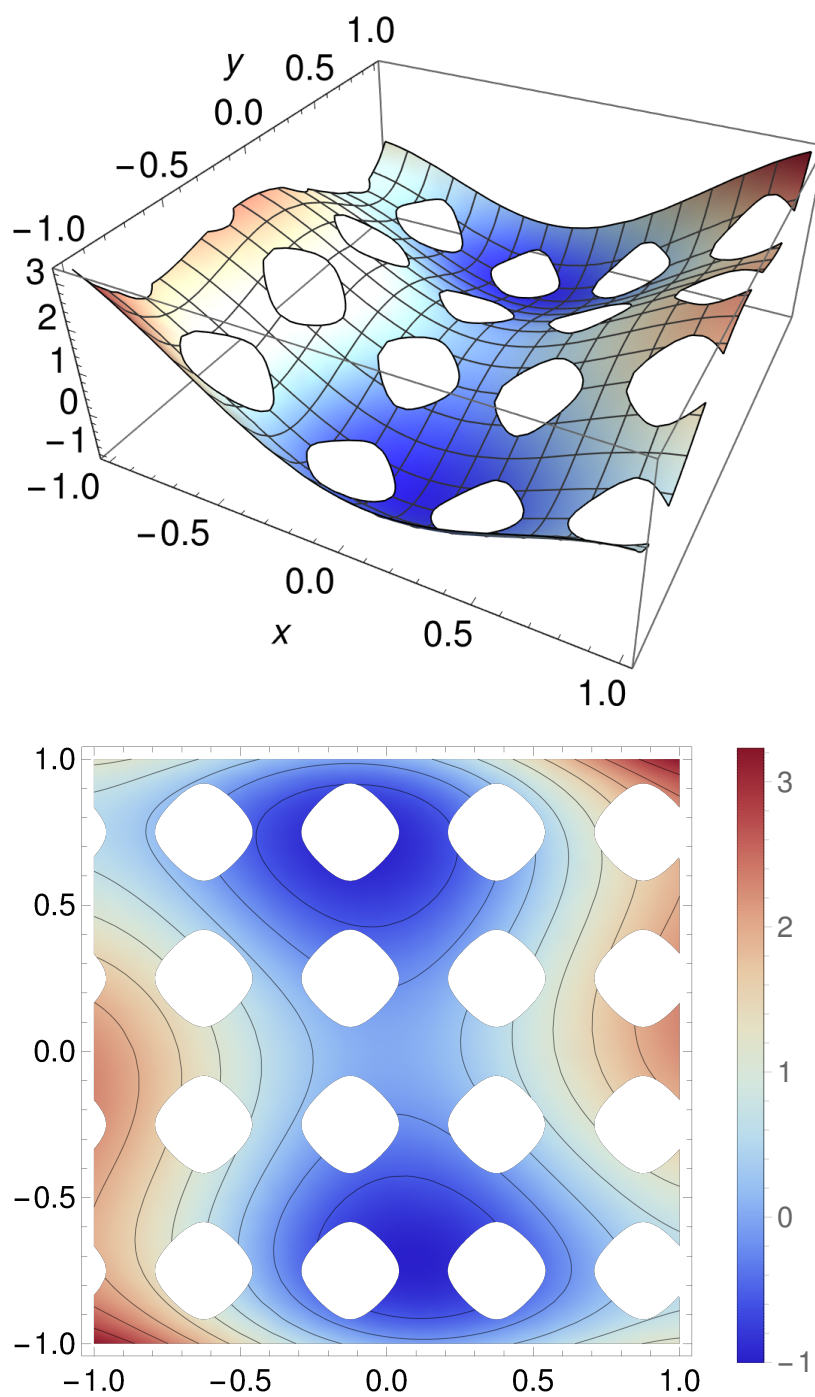


Figura A.7: Gráfica de la función de Gomez-Levy en \mathbb{R}^3 y como mapa de calor

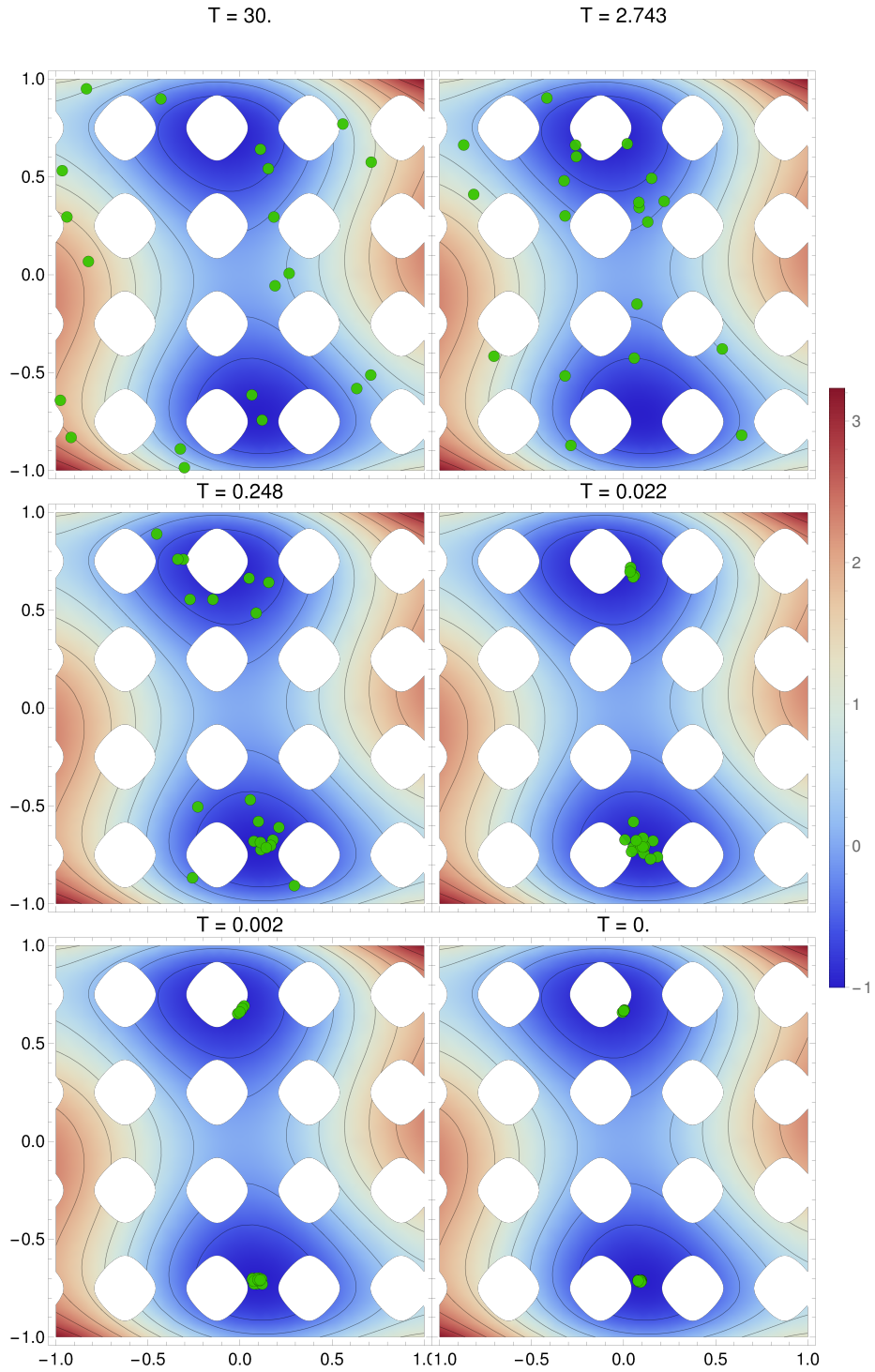


Figura A.8: Minimización de la función de Gomez-Levy mediante recocido simulado. Una animación del proceso puede encontrarse en <https://youtu.be/4Dy0Ww7t-g>.

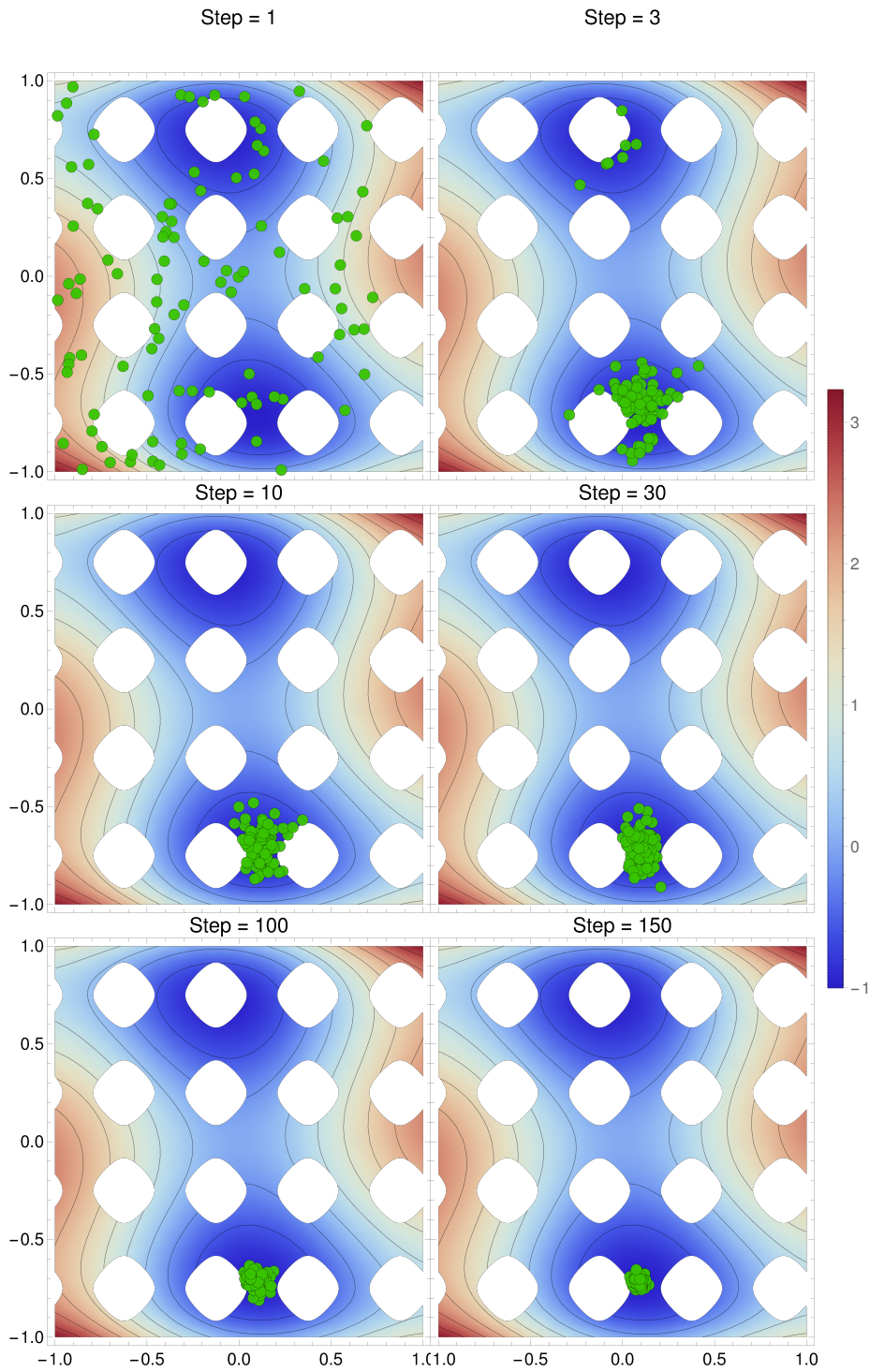


Figura A.9: Minimización de la función de Gomez-Levy mediante un algoritmo genético. Una animación del proceso puede encontrarse en <https://youtu.be/0uoD-S3dmVo>.

Anexo B: especificidades de los resultados

Coordenadas de las cadenas

Cadena aguja

La cadena poligonal de la figura 2.3 tal que se puede construir mediante las siguientes coordenadas:

$$\begin{aligned}\mathbf{v}_1 &= \mathbf{R}_x(-\pi/6)\mathbf{e}_x, \\ \mathbf{v}_2 &= \mathbf{e}_0, \\ \mathbf{v}_3 &= \mathbf{e}_x, \\ \mathbf{v}_4 &= \mathbf{e}_x + \mathbf{e}_z, \\ \mathbf{u} &= \mathbf{e}_x - \epsilon\mathbf{e}_y - \mathbf{v}_1, \\ \mathbf{v}_0 &= \mathbf{v}_1 + l\hat{\mathbf{u}}, \\ \mathbf{w} &= \epsilon\mathbf{e}_y - \mathbf{v}_4, \\ \mathbf{v}_5 &= \mathbf{v}_4 + l\hat{\mathbf{w}}.\end{aligned}\tag{B.1}$$

Con $\epsilon = 1/6$ y $\hat{\mathbf{x}}$ el vector unitario colineal a \mathbf{x} . Notemos que la cadena contará con la propiedad de que $l_1 = l_5 = l$ y $l_2 = l_3 = l_4 = 1$. Más aún, podemos hacer configuraciones genéricas de dicha cadena que sigan encerradas simplemente sumando un vector aleatorio de norma pequeña a cada vértice.

Cadena rectangular

La cadena de la figura 2.8 se puede generar con los siguientes vértices

$$\begin{aligned}
 \mathbf{v}_0 &= l\mathbf{e}_y + \epsilon\mathbf{e}_x, \\
 \mathbf{v}_1 &= \epsilon\mathbf{e}_z, \\
 \mathbf{v}_2 &= \mathbf{e}_x, \\
 \mathbf{v}_3 &= \mathbf{e}_x + \mathbf{e}_y, \\
 \mathbf{v}_4 &= \mathbf{e}_y + \epsilon\mathbf{e}_z, \\
 \mathbf{v}_5 &= \mathbf{e}_0, \\
 \mathbf{v}_6 &= l\mathbf{e}_x + \epsilon(\mathbf{e}_y + \mathbf{e}_z).
 \end{aligned} \tag{B.2}$$

Con $\epsilon = 0.1$. Esta cadena tiene la propiedad de que $l_1, l_6 \approx l$, $l_2, l_3, l_4 \approx 1$. De igual manera, los ángulos internos θ_i están cercanos a $\pi/2$, aunque no tienen exactamente dicho valor. En realidad, la condición de $l_2 = l_3 = l_4 = 1$ no se puede cumplir de forma simultánea a $\theta_i = \pi/2$, tal como es planteado en [24].

Si deseamos tener $\theta_i = \pi/2$ de forma estricta, entonces debemos poner

$$\begin{aligned}
 \mathbf{v}_2 &= \mathbf{e}_y, \\
 \mathbf{v}_3 &= \mathbf{e}_0, \\
 \mathbf{v}_4 &= \mathbf{e}_x, \\
 \mathbf{v}_1 &= \mathbf{v}_2 + \mathbf{R}_y(\alpha_1)\mathbf{e}_x, \\
 \mathbf{u} &= \mathbf{v}_1 - \mathbf{v}_2, \\
 \mathbf{v}_0 &= \mathbf{v}_1 - l\mathbf{R}_{\hat{\mathbf{u}}}(\alpha_2)\mathbf{e}_y, \\
 \mathbf{v}_5 &= \mathbf{v}_4 - \mathbf{R}_x(\alpha_1)\mathbf{e}_y, \\
 \mathbf{w} &= \mathbf{v}_5 - \mathbf{v}_4, \\
 \mathbf{v}_6 &= \mathbf{v}_5 - l\mathbf{R}_{\hat{\mathbf{w}}}(\alpha_2)\mathbf{e}_x,
 \end{aligned} \tag{B.3}$$

Con $\alpha_1 = -0.012566$ y $\alpha_2 = 0.131947$. Cabe mencionar que la diferencia entre ambas cadenas es apenas perceptible a simple vista.

Nudo Trefoil

La parametrización del nudo trefoil simple está dada por las coordenadas paramétricas

$$\begin{cases} x(t) = \cos(t) + 2 \cos 2t, \\ y(t) = \sin(t) - 2 \sin 2t, \\ z(t) = 2 \sin 3t. \end{cases} \tag{B.4}$$

Para $t \in [0, 2\pi]$. La versión polinomial de dicho nudo está dada por

$$\begin{cases} x(t) = t^3 - 3t, \\ y(t) = t^4 - 4t^2, \\ z(t) = \frac{t^5}{5} - 2t. \end{cases} \quad (\text{B.5})$$

La parte principal del nudo puede obtenerse con $t \in [-2.1, 2.1]$. Para discretizar cualquiera de estos nudos, basta tomar una partición regular del intervalo de t y tomar esos puntos como vértices. Algo importante de diferenciar es que los vértices quedan espaciados de forma casi equidistante en la versión simple, mientras que en la versión polinomial los vértices se concentran en la parte central del nudo. Dicho comportamiento es esperado y esencial para el buen funcionamiento de los algoritmos basados en descenso de gradiente.

Nudo Figura 8

La parametrización del nudo de figura 8 se puede obtener como:

$$\begin{cases} x(t) = 3 \cos t + 5 \cos 3t, \\ y(t) = 3 \sin t + 5 \sin 3t, \\ z(t) = \sin\left(\frac{5t}{2}\right) \sin 3t + \sin 4t - \sin 6t. \end{cases} \quad (\text{B.6})$$

Con $t \in [0, 2\pi]$. Otra versión, que presenta el nudo más pequeño en comparación con toda la curva, se puede parametrizar como:

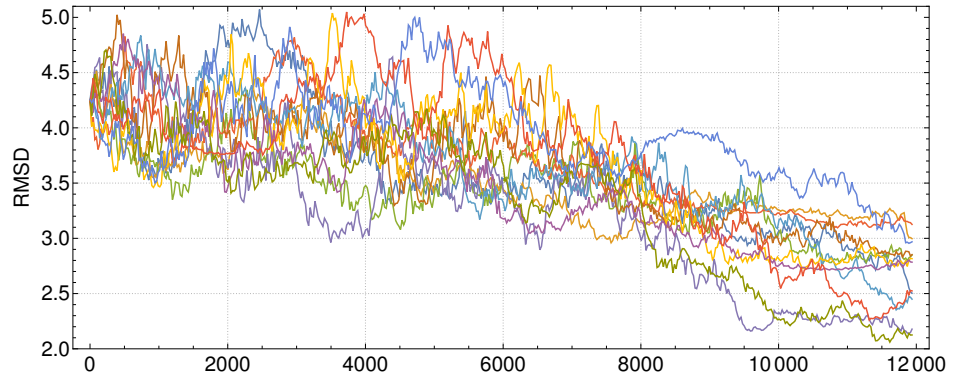
$$\begin{cases} x(t) = \sin\left(t + \frac{t}{10}\right), \\ y(t) = \sin t \cos \frac{t}{2}, \\ z(t) = \frac{\sin 2t \cos \frac{t}{2}}{2}. \end{cases} \quad (\text{B.7})$$

Nuevamente, para $t \in [0, 2\pi]$. Una versión polinomial se puede definir como

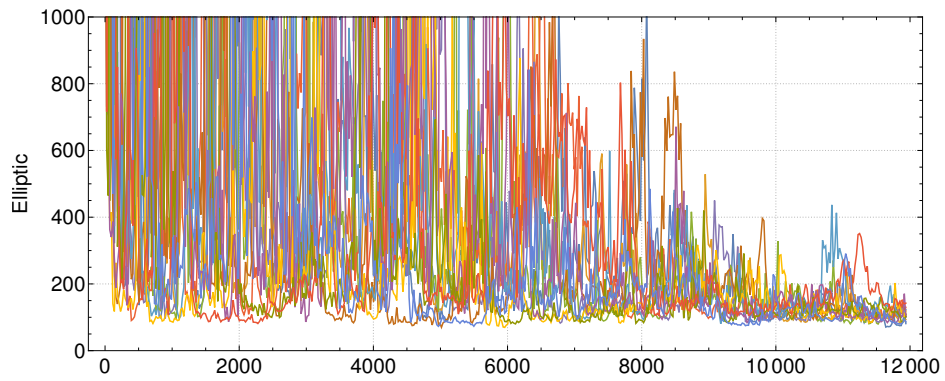
$$\begin{cases} x(t) = \frac{2t}{5}(t^2 - 7)(t^2 - 10), \\ y(t) = t^4 - 13t^2, \\ z(t) = \frac{t}{10}(t^2 - 4)(t^2 - 9)(t^2 - 12). \end{cases} \quad (\text{B.8})$$

Con $t \in [-3.7, 3.7]$.

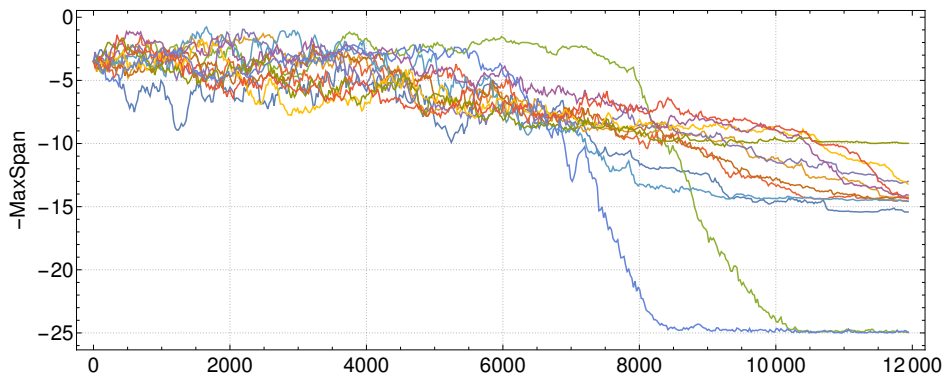
Simulaciones



(a)



(b)



(c)

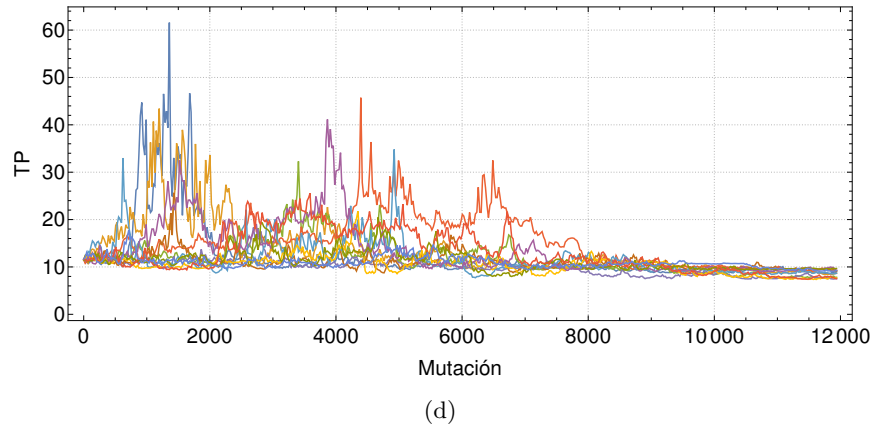
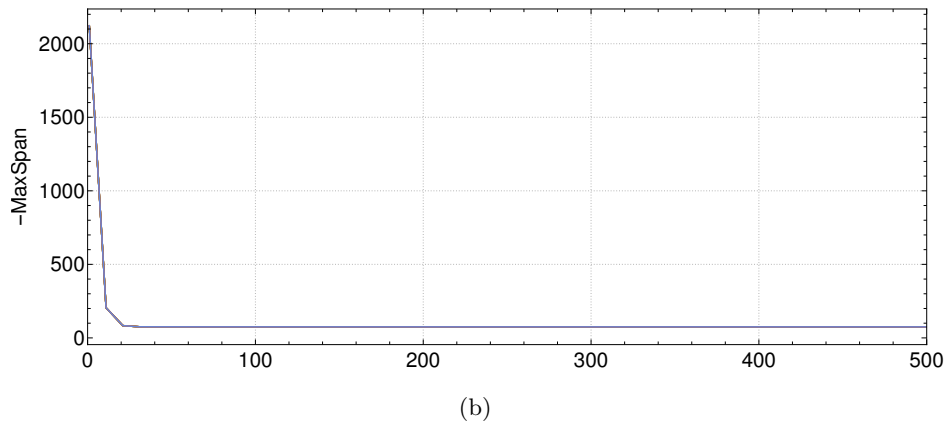
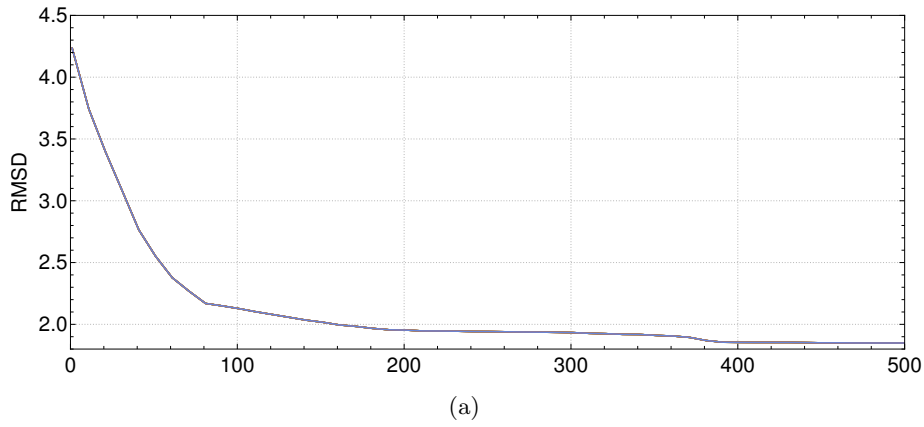
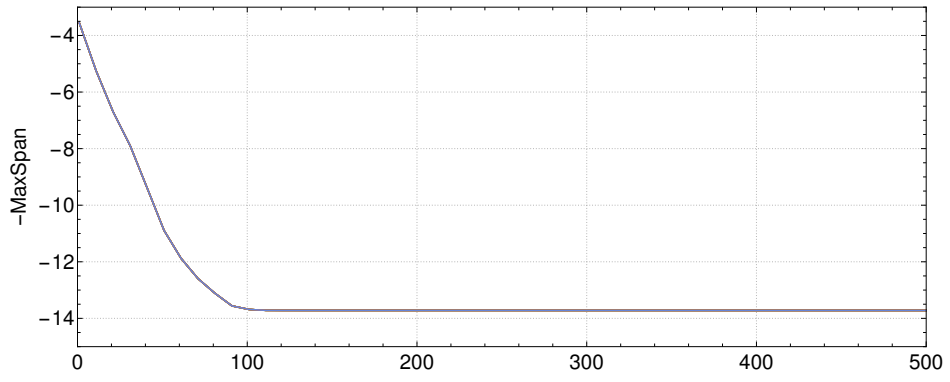
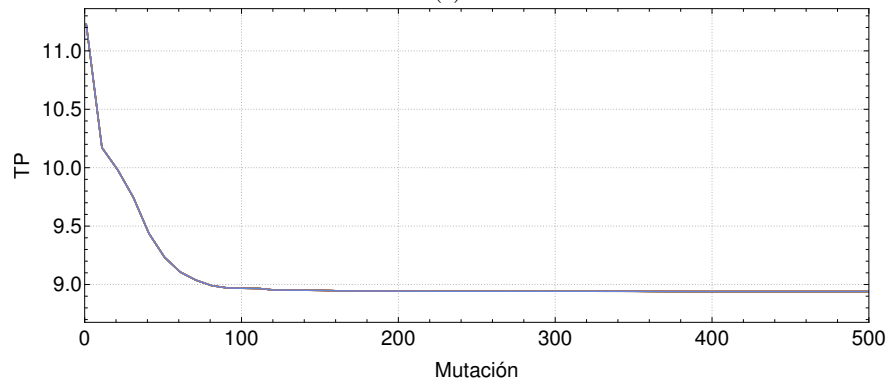


Figura B.1: Trayectorias de minimización mediante annealing para la cadena aguja para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) -MaxSpan, y d) TP



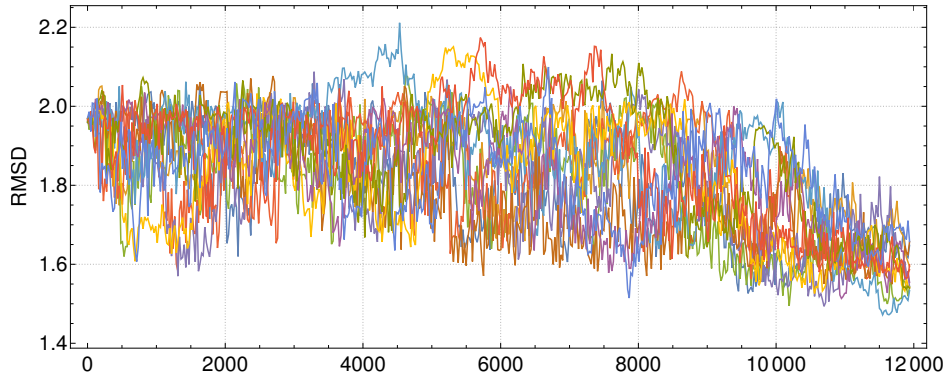


(c)

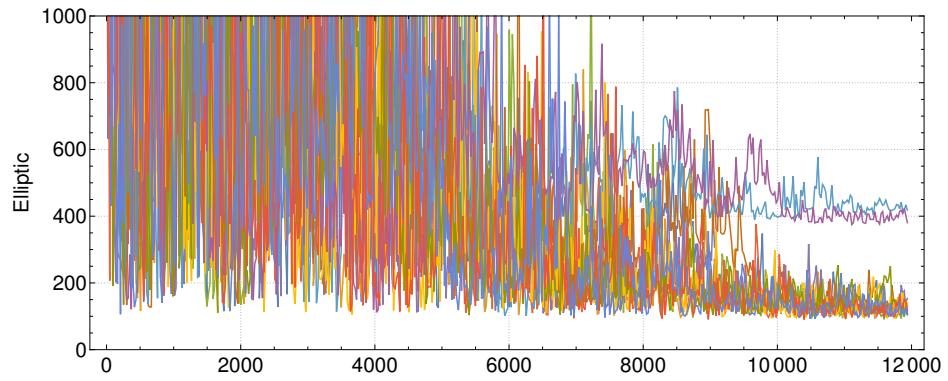


(d)

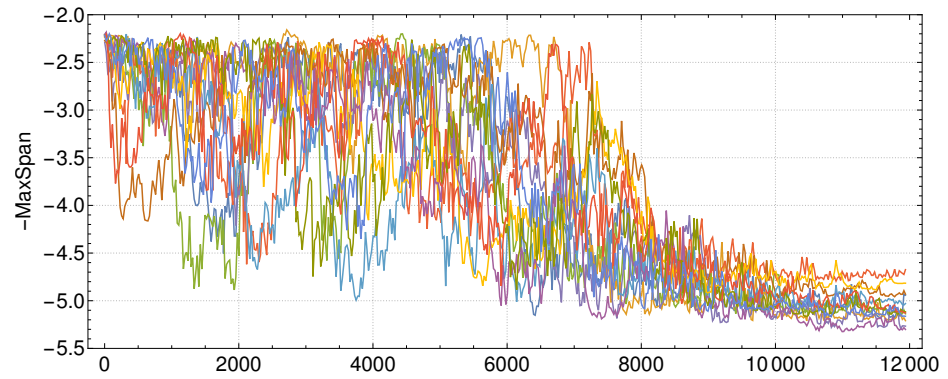
Figura B.2: Trayectorias de minimización mediante genético para la cadena aguja para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) $-\text{MaxSpan}$, y d) TP



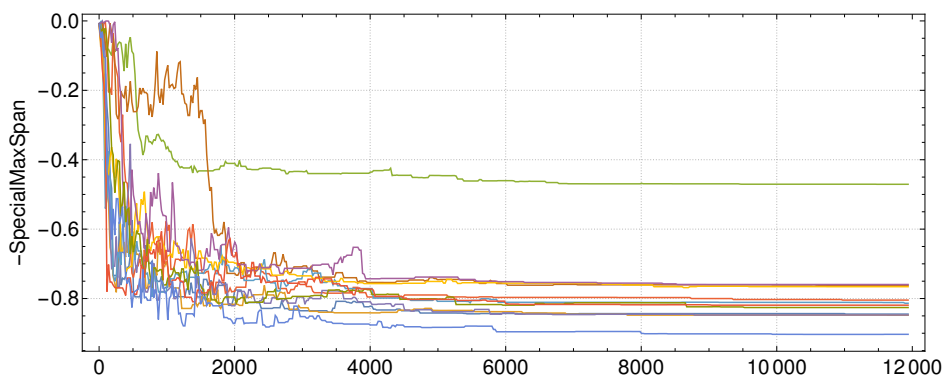
(a)



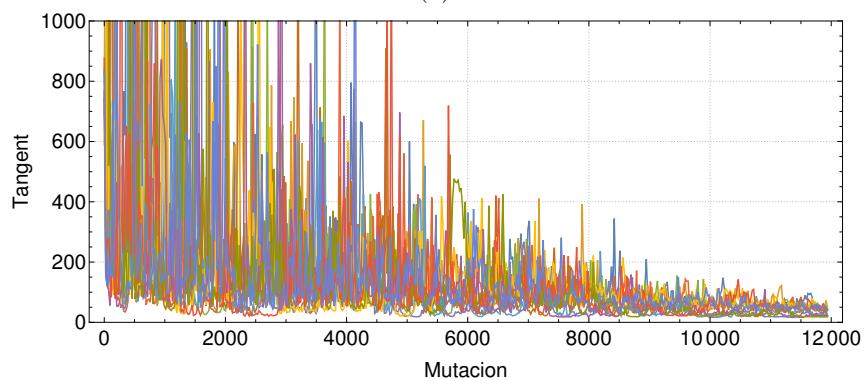
(b)



(c)

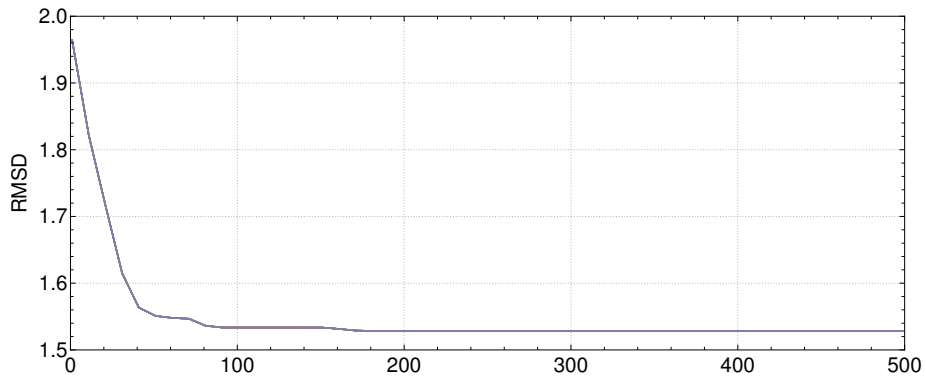


(d)

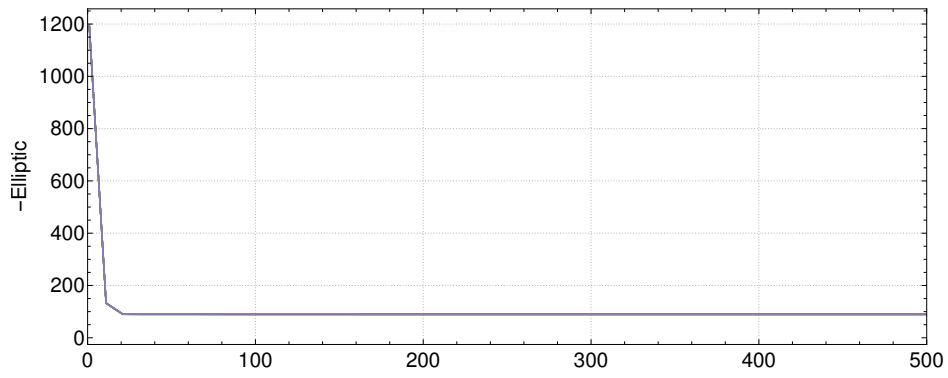


(e)

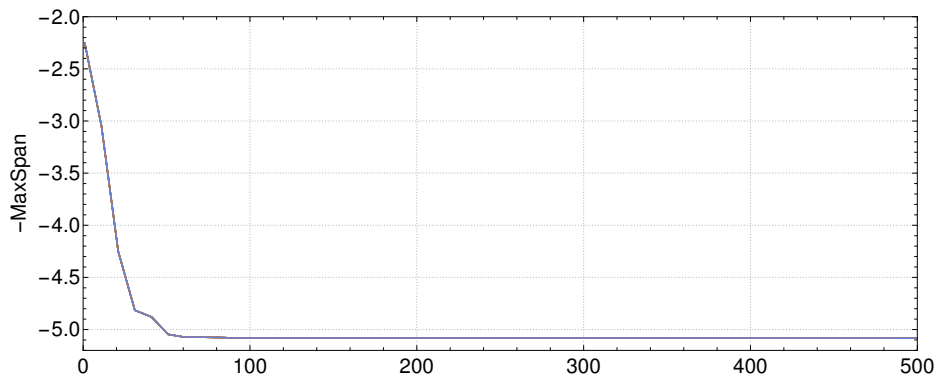
Figura B.3: Trayectorias de minimización mediante annealing para la cadena rectangular para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) $-MaxSpan$, d) $-SpecialMaxSpan$ y e) TP



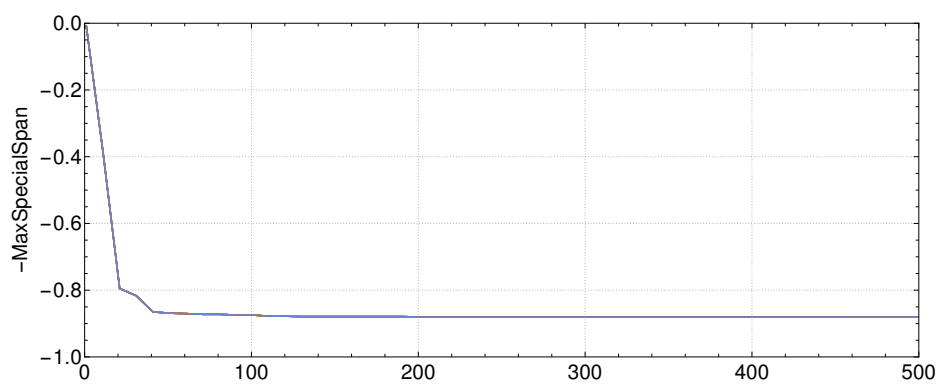
(a)



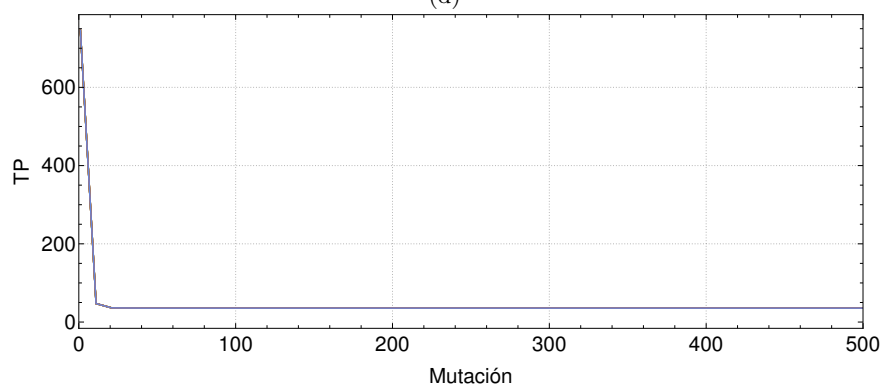
(b)



(c)



(d)



(e)

Figura B.4: Trayectorias de minimización mediante algoritmo genético para la cadena rectangular para diferentes energías. a) RMSD a la configuración plana, b) Elliptic, c) $-\text{MaxSpan}$, d) $-\text{SpecialMaxSpan}$ y e) TP

En las tablas B.1 y B.2 se encuentra una relación con videos de simulaciones realizadas.

Cadena	Algoritmo	Energía	Video
Agujas, $l = 2$	Genético	Elíptica	https://youtu.be/3lgIOWQWSVc
		Tangente	https://youtu.be/nmrU1Moenz0
		MaxSpan	https://youtu.be/k9iLcnR9Fu0
		RMSD a conf. recta	https://youtu.be/kqVwgHsSDmE
	Annealing	Elíptica	https://youtu.be/3ADu_5UFCPI
		Tangente	https://youtu.be/gVAPq01WwjA
		MaxSpan	https://youtu.be/ZcK-fvOBBqA
		RMSD a conf. recta	https://youtu.be/h8miz9Tp_gw
	FGD	Tangente	https://youtu.be/8X9L1-772Do
	FGD, alta resolución	Tangente	https://youtu.be/TDum1Vad1Fo
Agujas, $l = 2.589$	Annealing	MaxSpan	https://youtu.be/R3W1PB20feU
Rectangular, $l = 1.15$	Genético	Elíptica	https://youtu.be/SpHYMZnHrJY
		Tangente	https://youtu.be/Ang-mB4XqHA
		MaxSpan	https://youtu.be/6QKQJa9UdiM
		SpecialSpan	https://youtu.be/HGQcXYC-atA
		RMSD a conf. aplanada	https://youtu.be/ak60U92IULI
	Annealing	Elíptica	https://youtu.be/8xfDZyNZd2Q
		Tangente	https://youtu.be/821QpwHFPIU
		MaxSpan	https://youtu.be/rRGGADChbrE
		SpecialSpan	https://youtu.be/kaa6iJENRJO
		RMSD a conf. aplanada	https://youtu.be/_2WlxcbWH_c
Rectangular, $l = 1.318$	Annealing	SpecialSpan	https://youtu.be/R3W1PB20feU

Tabla B.1: Videos para simulaciones de cadenas poligonales.

Trefoil $n = 50$	FGD	Tangente	https://youtu.be/mG-rM0nIba4
Trefoil Polinomial $n = 60$			https://youtu.be/mrtKRIAimdc
Figura 8, $n = 60$			https://youtu.be/fhJmOW40QN0
Figura 8 Apretada, $n = 60$			https://youtu.be/V_fk3WsEzwe
Figura 8 Polinomial, $n = 70$			https://youtu.be/pIRIqLCPVKQ

Tabla B.2: Videos para simulaciones de curvas suaves.

Para evitar la necesidad de ir a revisar los videos, anexamos aquí algunos cuadros de cada video.

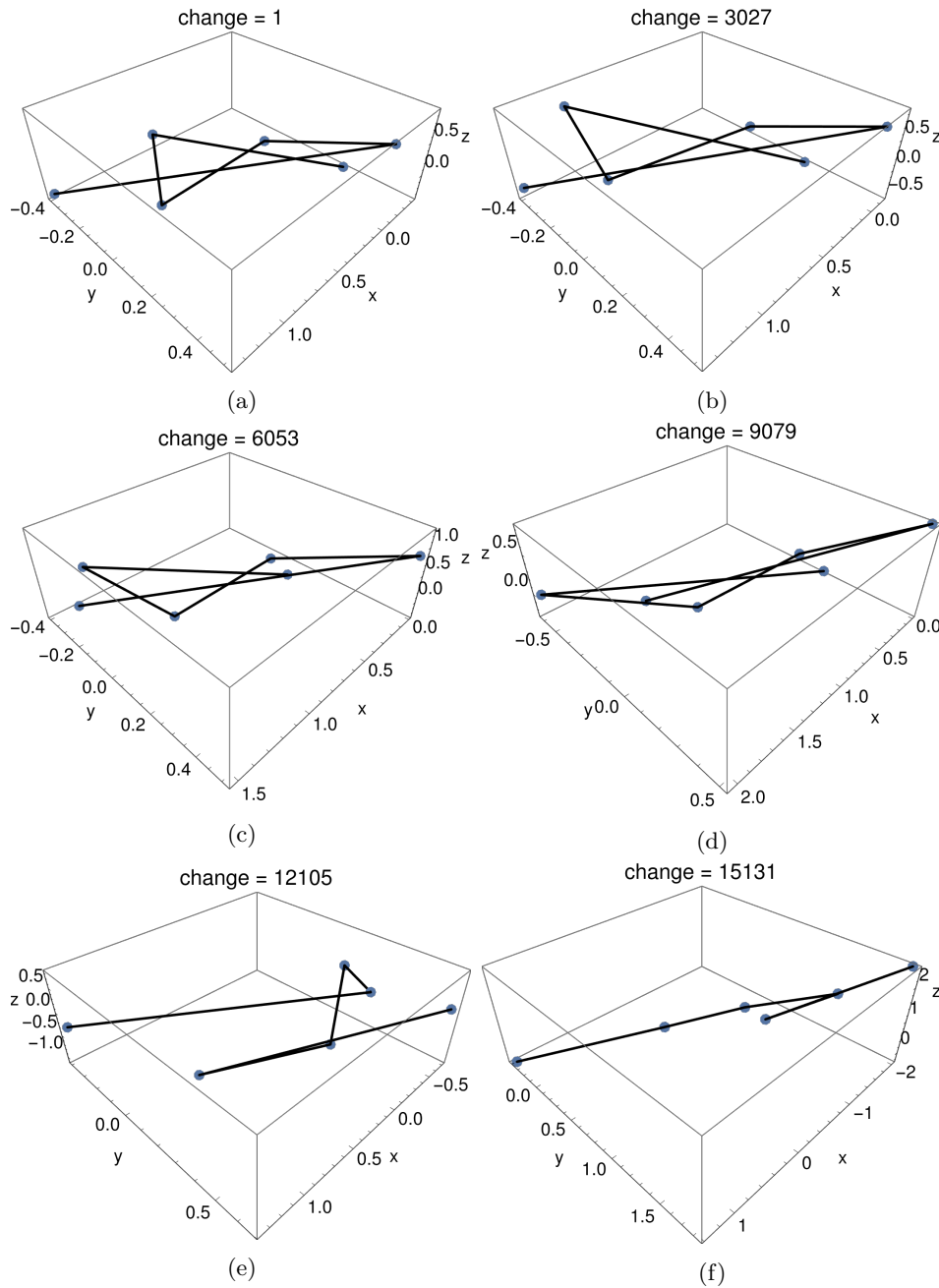


Figura B.5: Simulación de la cadena aguja, minimizando la energía del MaxSpan (ecuación 3.42) mediante Annealing.

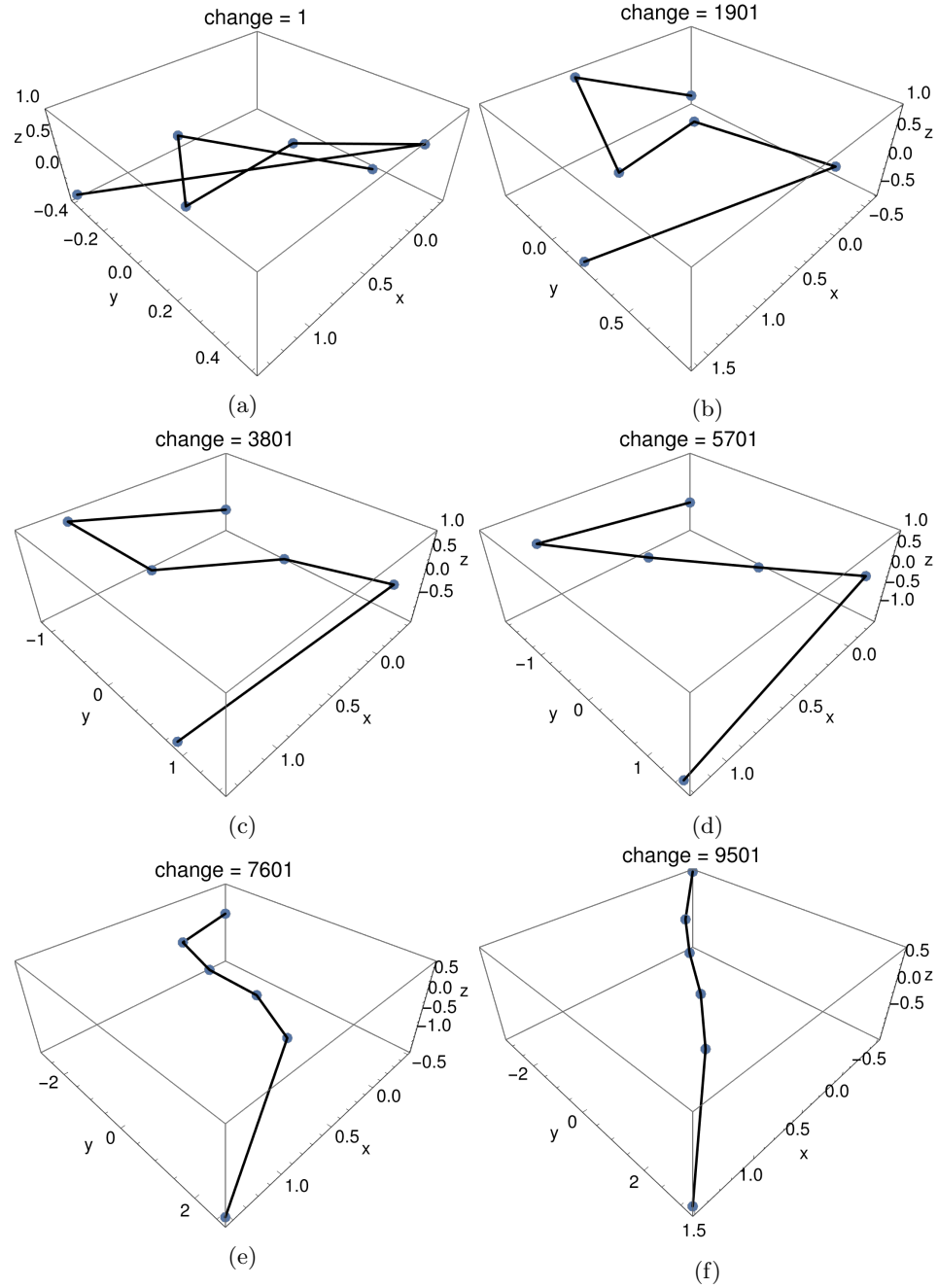


Figura B.6: Simulación de la cadena aguja, minimizando la energía de punto tangente (ecuación 3.32) mediante el algoritmo FGD.

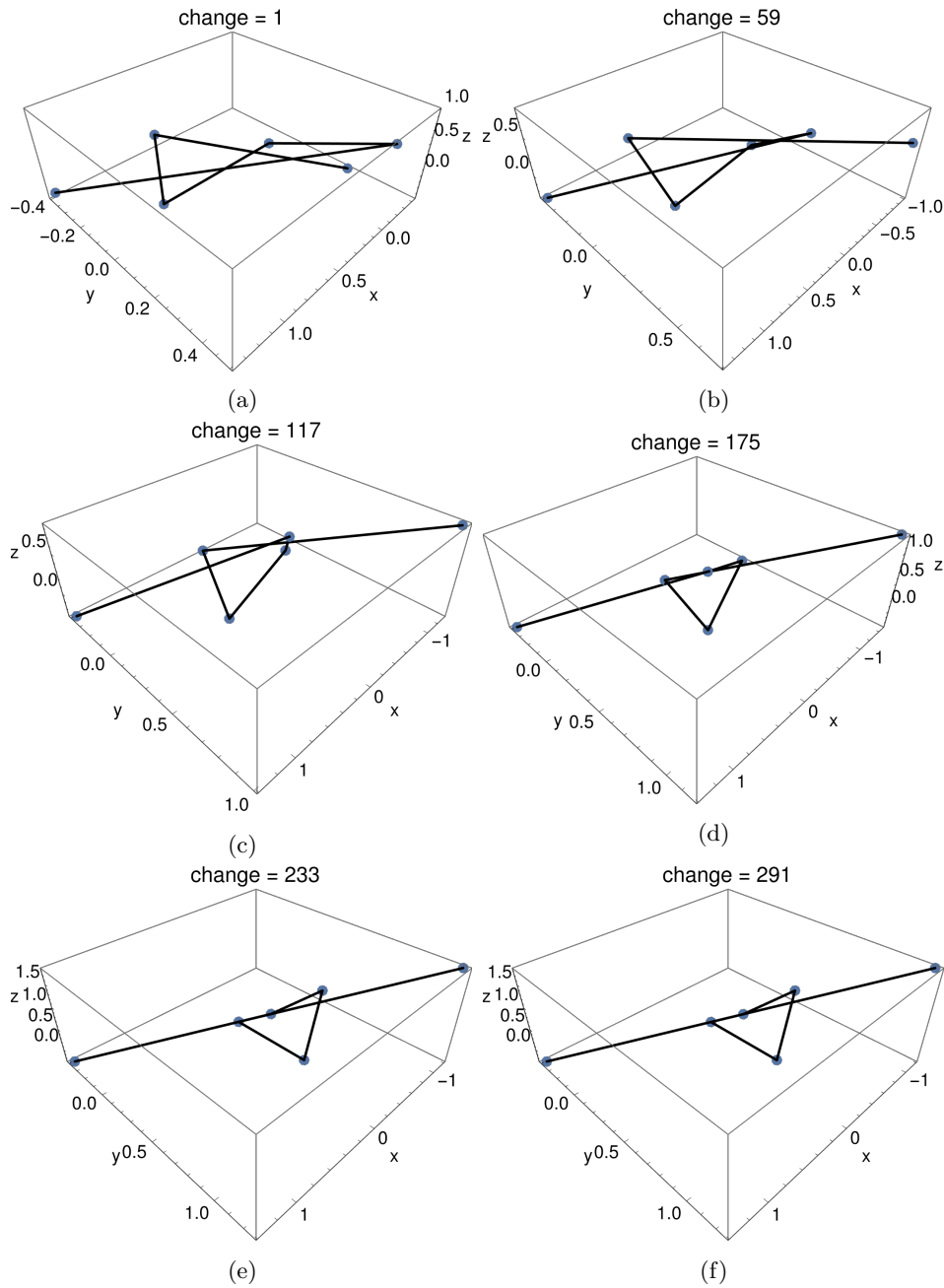


Figura B.7: Simulación de la cadena aguja, minimizando la energía del MaxSpan (ecuación 3.42) mediante el algoritmo genético.

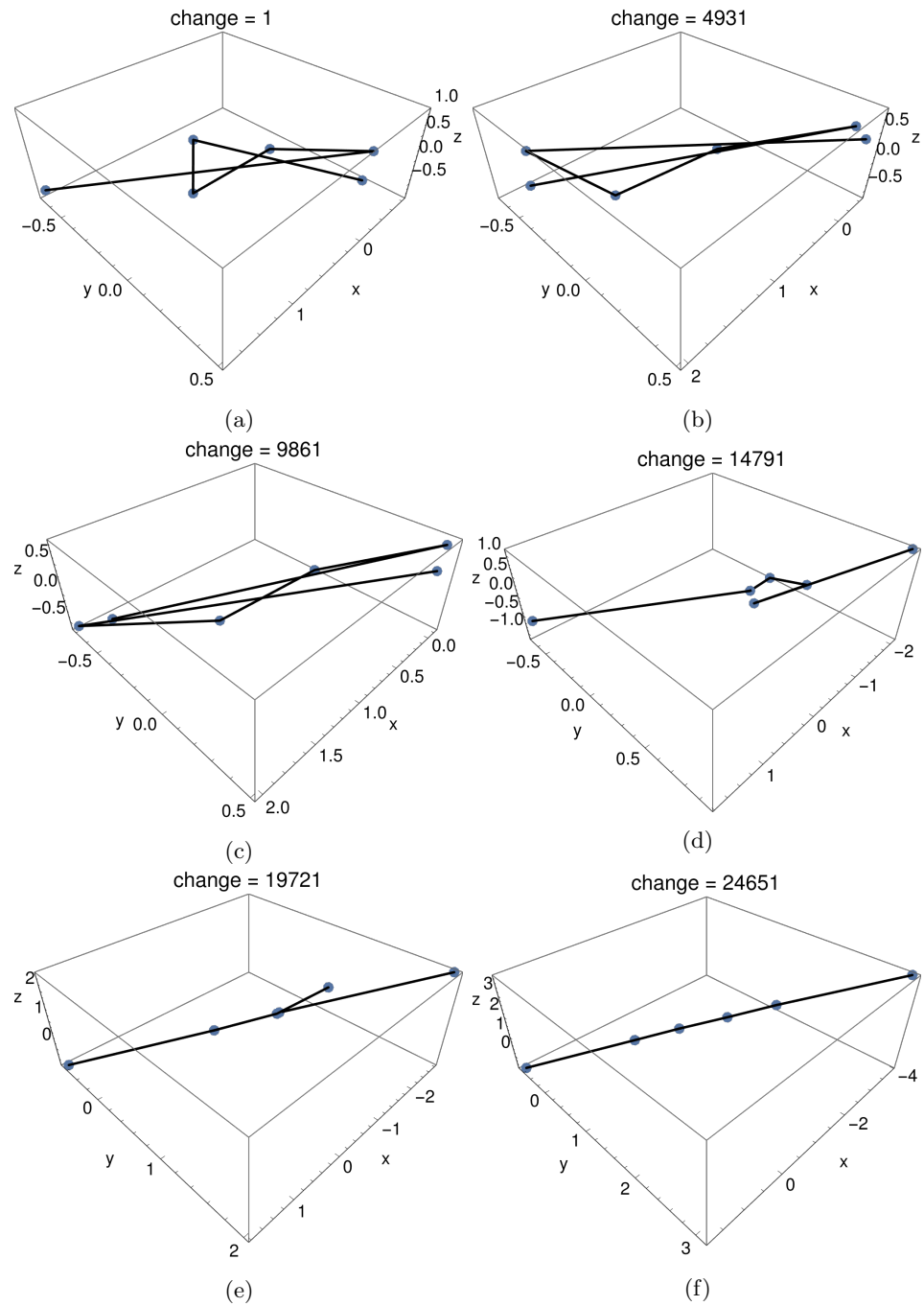


Figura B.8: Simulación de la aguja rectangular con $l = 2.589$, minimizando la energía del MaxSpan (ecuación 3.42) mediante el algoritmo Annealing con hiperparámetros modificados. Se tomó una temperatura inicial de 15 (en unidades de la energía inicial) y una final de 10^5 , con un número máximo de 300,000 iteraciones.

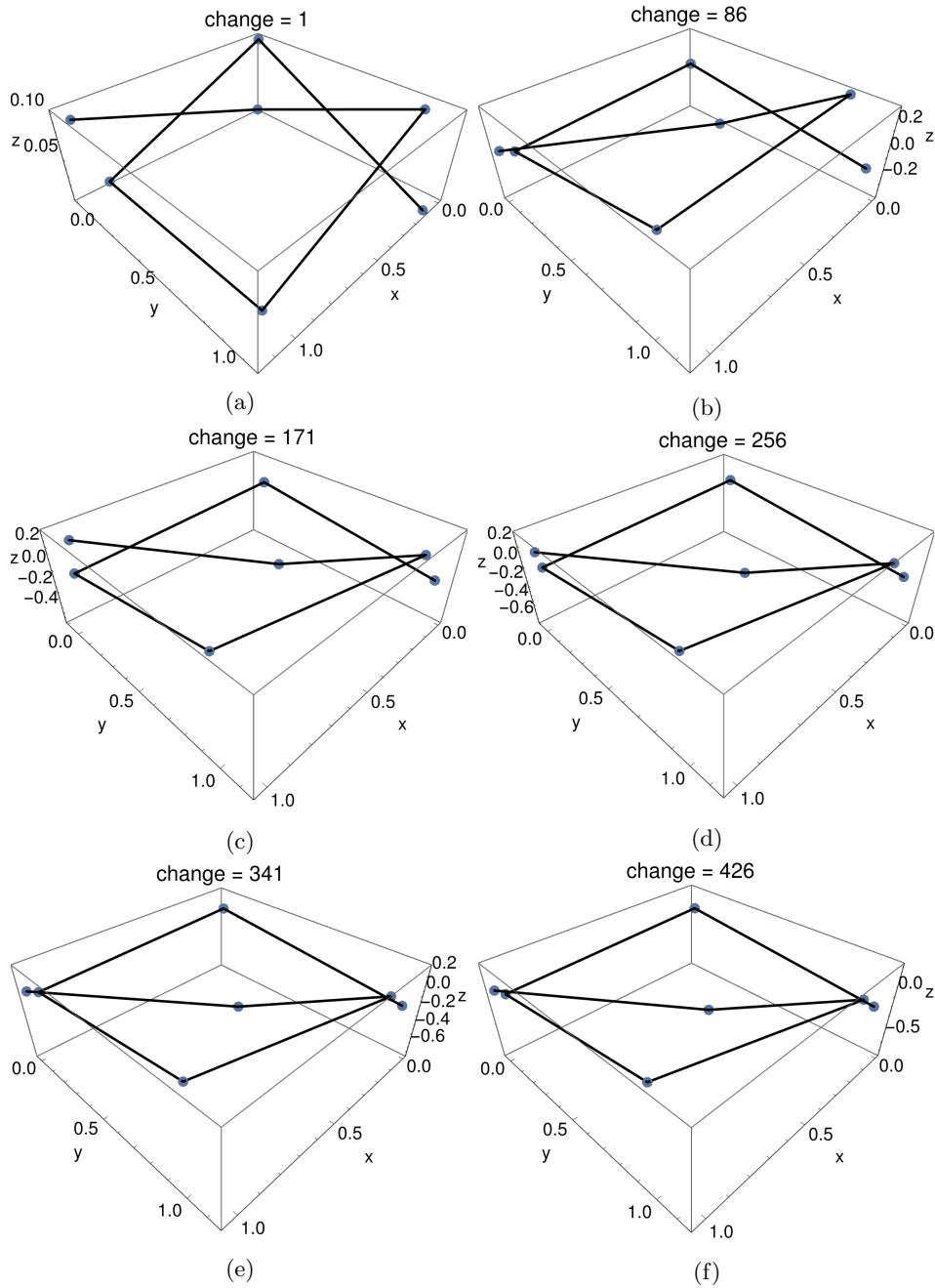


Figura B.9: Simulación de la cadena rectangular, minimizando la energía del SpecialSpan (ecuación 3.43) mediante el algoritmo genético.

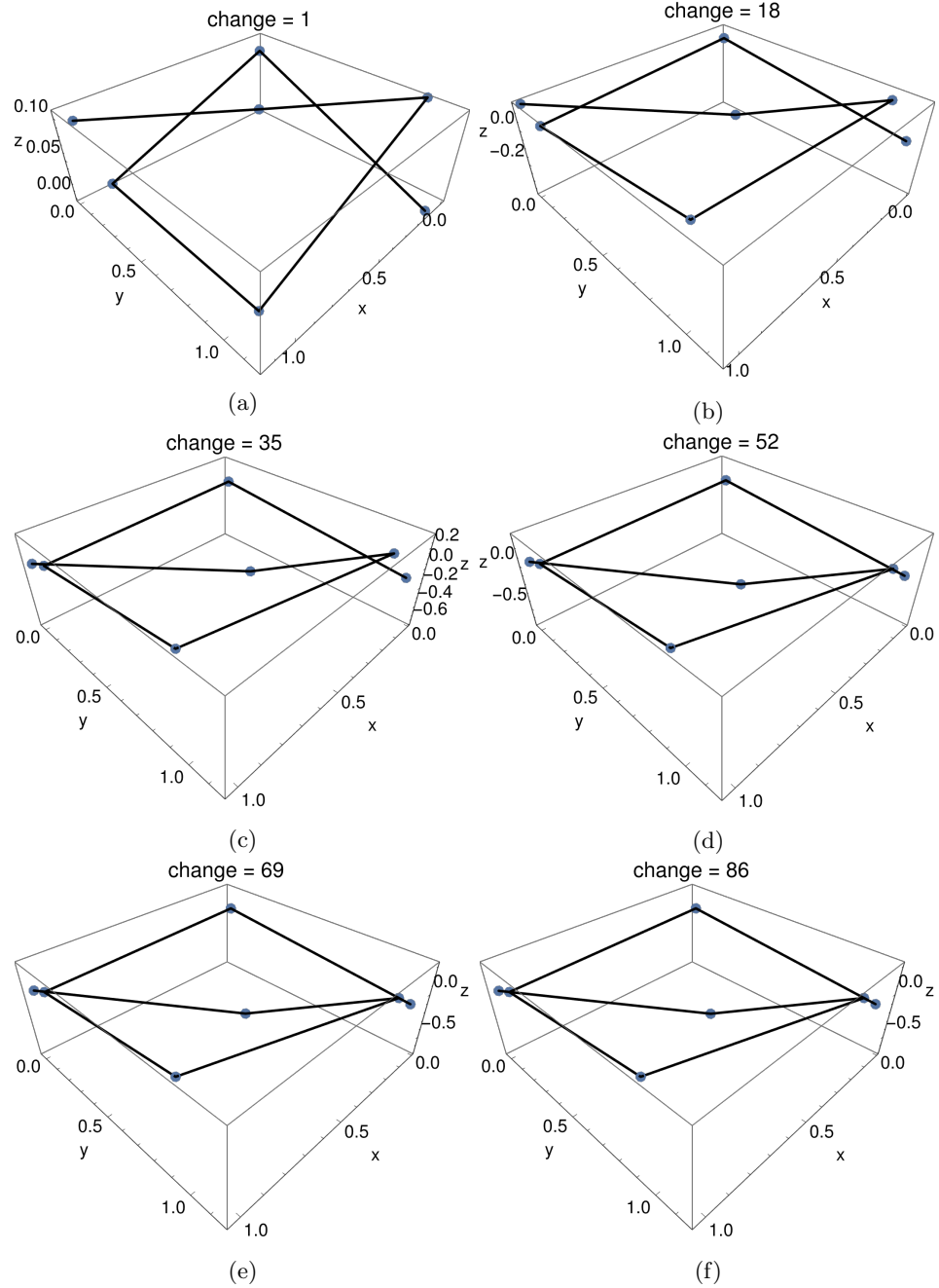


Figura B.10: Simulación de la cadena rectangular, minimizando la energía del SpecialSpan (ecuación 3.32) mediante el algoritmo genético.

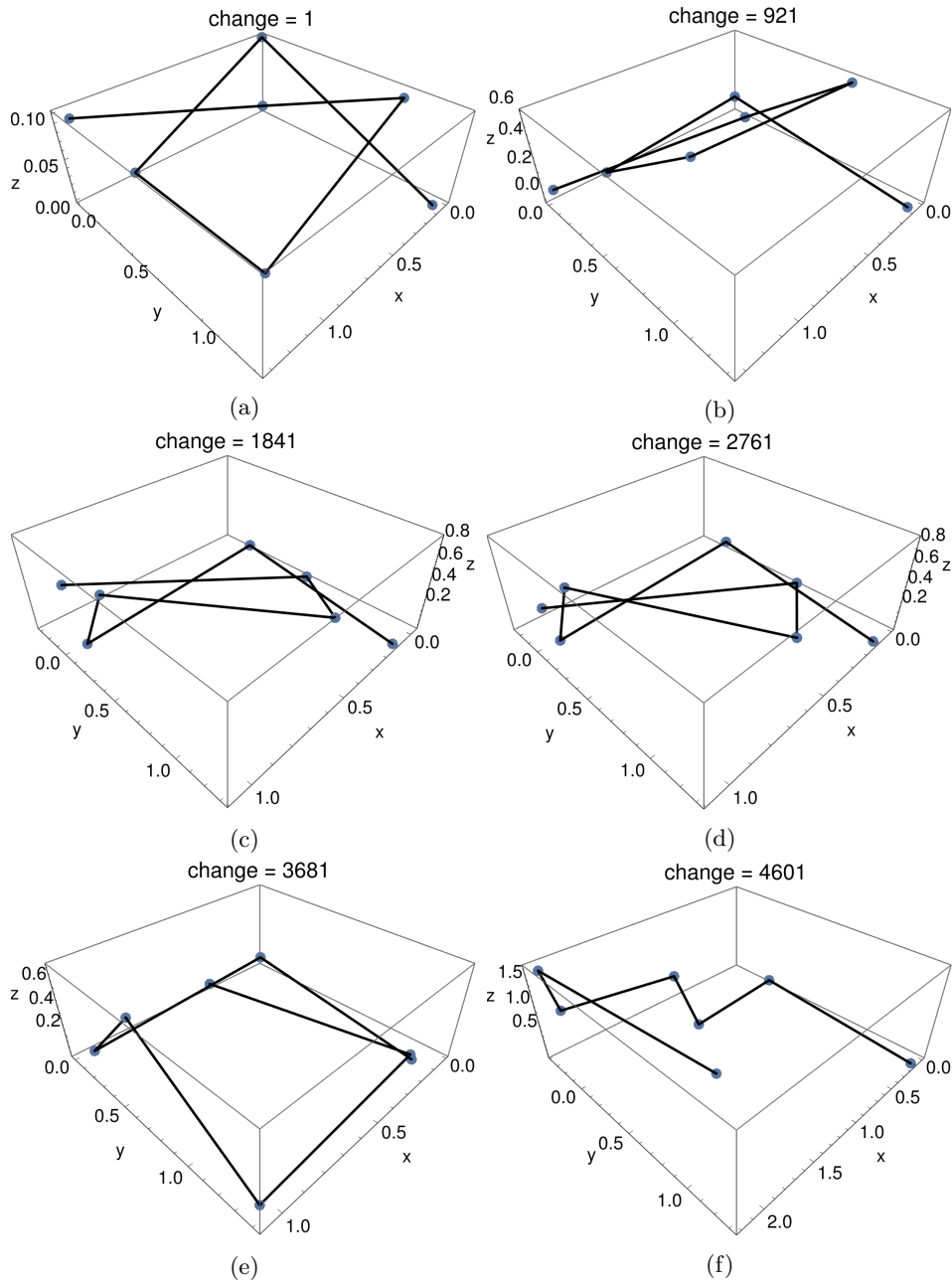


Figura B.11: Simulación de la cadena rectangular con $l = 1.319$, minimizando la energía del SpecialSpan (ecuación 3.43) mediante el algoritmo Annealing con hiperparámetros modificados. Se tomó una temperatura inicial de 15 (en unidades de la energía inicial) y una final de 10^5 , con un número máximo de 300,000 iteraciones.

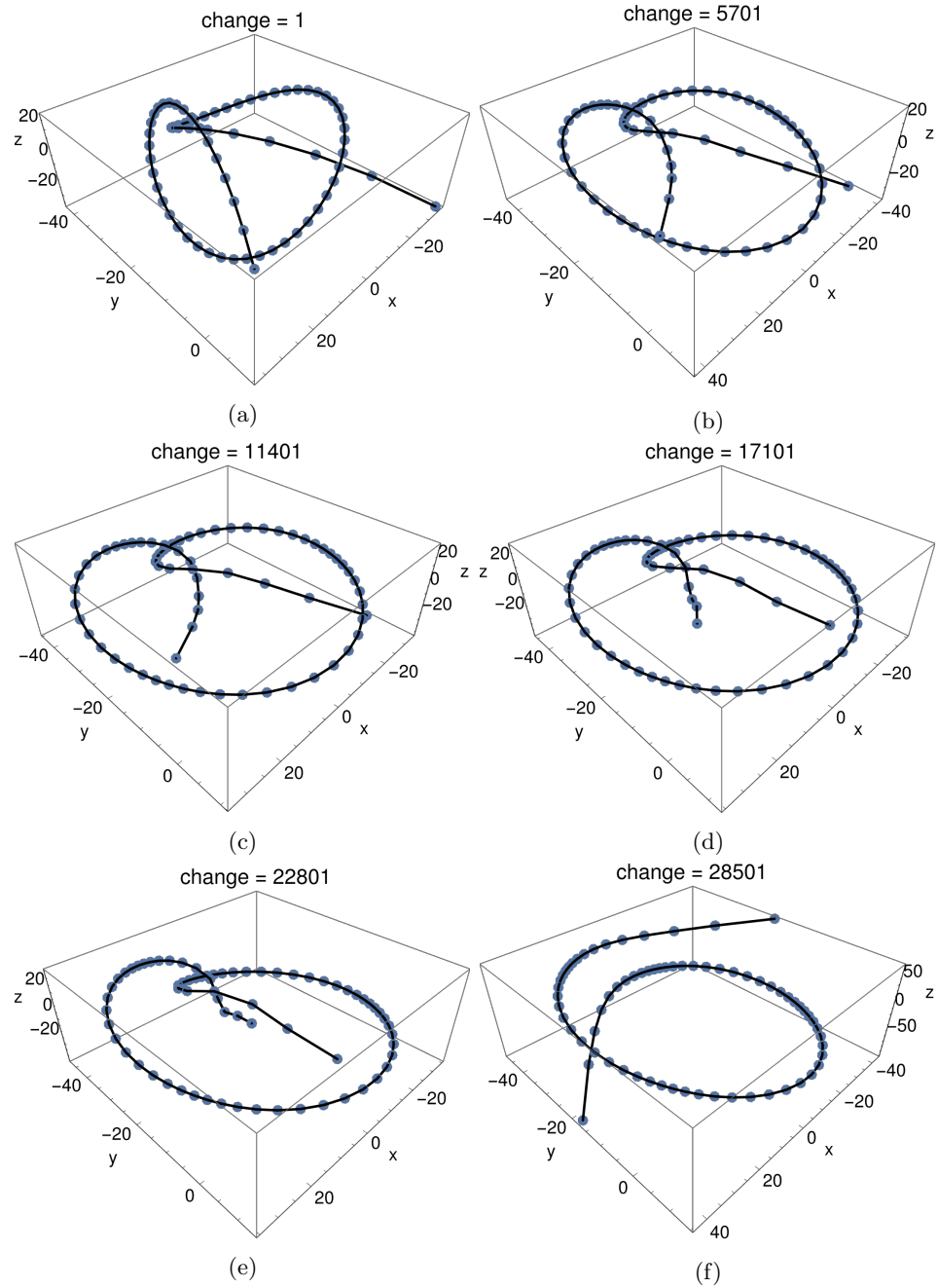


Figura B.12: Simulación del nudo figura 8 en versión polinomial, minimizando la energía de punto tangente mediante el algoritmo FGD.

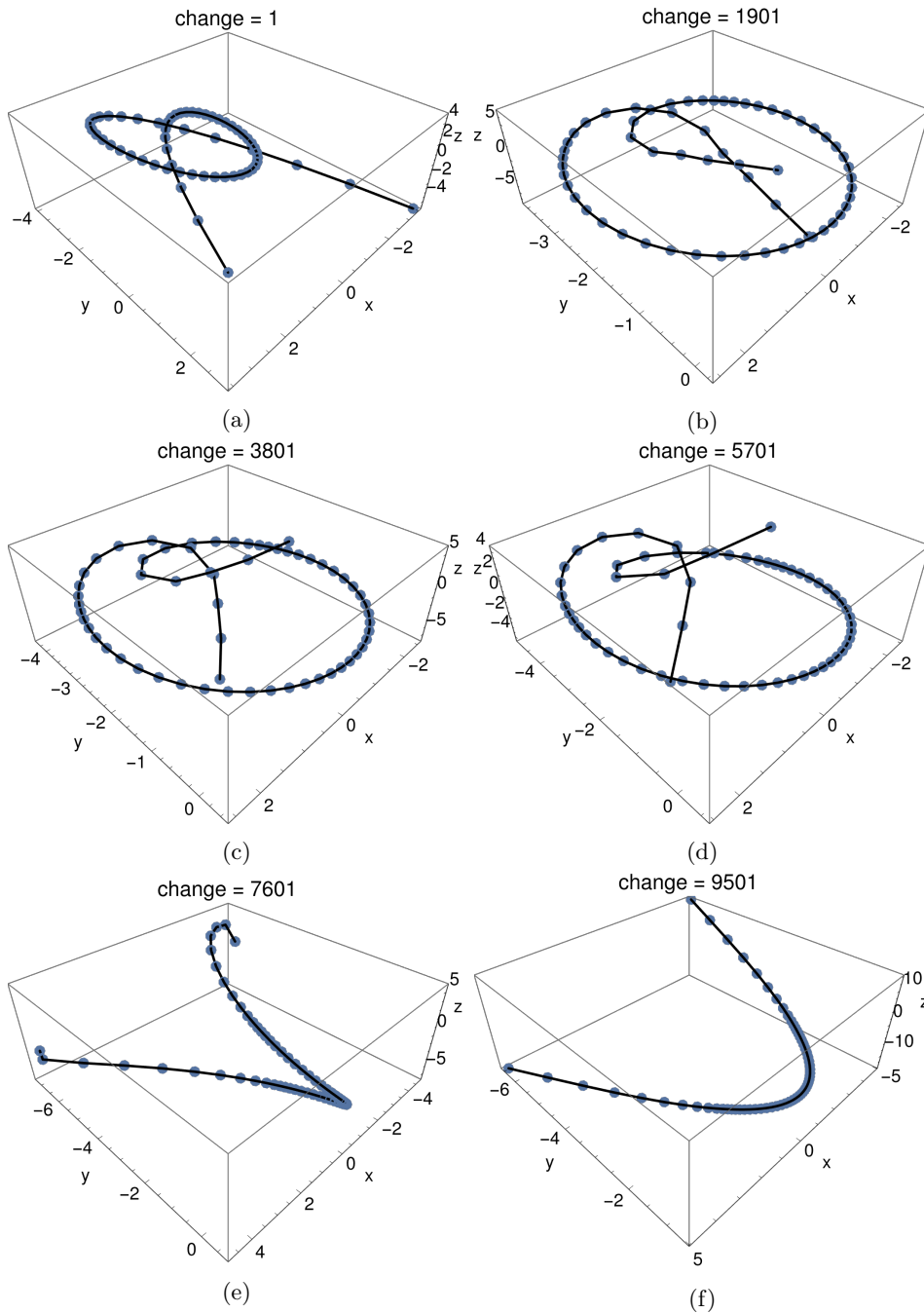


Figura B.13: Simulación del nudo trefoil en versión polinomial, minimizando la energía de punto tangente (ecuación 3.32) mediante el algoritmo FGD.