



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

CÁLCULO DE VOLÚMENES DE POLITOPOS EN ESPACIOS DE ALTAS DIMENSIONES

Constanza Lila Dieci Seré

Tutores:

Héctor Cancela

Eduardo Fernández

Proyecto de Grado en Ingeniería en Computación

Facultad de Ingeniería

Universidad de la República

Montevideo – Uruguay

Marzo 2022

Resumen

En este trabajo se estudia la aplicabilidad de los métodos de Monte Carlo para el estudio de las redes metabólicas de microorganismos como la *Escherichia coli*. En particular, se busca estimar el volumen del espacio de soluciones determinado por las posibles configuraciones de las redes mencionadas. Este problema surge en la literatura del área, donde se utilizan modelos lineales a partir de la información genómica y de restricciones adicionales, cuyos espacios de soluciones son politopos expresados como la intersección de un conjunto finito de semiespacios cerrados ($H - \text{Politopos}$). En un trabajo de maestría previo realizado en Udelar se abordó este mismo problema, llegando a estimar volúmenes con una cantidad limitada de dimensiones. Por lo tanto, el desafío pendiente que se planteó para abordar en este trabajo fue el estudio de algoritmos que permitieran aproximar el volumen utilizando este tipo de representación para espacios con mayor dimensión que la que fue alcanzada en la tesis mencionada.

En el presente trabajo se realizó un relevamiento de estado del arte. A partir de ese estudio, se seleccionaron e implementaron tres métodos basados en Monte Carlo, que fueron vistos en trabajos relacionados al estudio específico de redes metabólicas. Por otra parte, se estudiaron y utilizaron los algoritmos, también basados en Monte Carlo, implementados en la librería de código abierto VolEsti. A su vez se estudiaron herramientas de software que calculan el volumen exacto de politopos regulares, para poder verificar la precisión de los resultados obtenidos con los otros algoritmos. A partir del estudio de estas herramientas pudimos recopilar un conjunto de politopos básico, para el cual se conoce su volumen. Los algoritmos basados en Monte Carlo que estudiamos fueron aplicados a este conjunto básico de politopos, con el fin de evaluar su desempeño antes de pasar a la etapa de aplicación de los mismos en los modelos metabólicos. El politopo de mayor dimensión que utilizamos en estas pruebas fue el politopo de Birkhoff de dimensión 81. En esta etapa del trabajo, verificamos que los resultados se aproximaban al volumen exacto con un error relativo porcentual aceptable en la mayoría de los politopos (incluso en los de mayor dimensión), pero hubo un subconjunto para los cuales los resultados no fueron buenos. Luego de estudiar las características de este subconjunto, vimos que se trataba de politopos con una forma alargada, es decir que tiene subespacios que se encuentran mucho más acotados que otros en determinadas direcciones. Obtener dichos resultados nos permitió verificar empíricamente lo que encontramos que decía la literatura con respecto a que los métodos estudiados no funcionan bien con politopos de estas características.

Finalmente, seleccionamos un subconjunto de los algoritmos estudiados para ser aplicados en dos modelos que representan la red metabólica del organismo unicelular *Escherichia coli* (modelo compacto *E.coli* y modelo núcleo *E.coli*). En este punto, cabe destacar que la forma característica de estos modelos es similar a la de los modelos básicos para los cuales no se obtuvieron buenos resultados en la etapa anterior. Esto se debe a que la manera de modelar condiciones ambientales o biológicas que representan características del comportamiento del metabolismo de los microorganismos es utilizar restricciones muy acotadas en relación a las demás. Al igual que en la etapa anterior del trabajo, en una primera instancia no obtuvimos buenos resultados, pero luego de realizar un ajuste al modelo, pudimos seleccionar los dos algoritmos que devolvieron mejores resultados. A partir de estos resultados aplicamos el método de Bland-Altman que es utilizado para determinar si 2 métodos diferentes que miden la misma propiedad concuerdan lo suficiente para que se pueda decir que son intercambiables. A los efectos de este trabajo considerando como indicadores el error relativo calculado a partir de los politopos para los cuales pudimos calcular el volumen exacto y el resultado del análisis mediante el método Bland-Altman concluimos que la aproximación del volumen que obtuvimos para el modelo compacto *E.coli* fue aceptable; mientras que para el caso del modelo núcleo *E.Coli*, de mayores dimensiones, no fue posible alcanzar una aproximación de buena calidad, mostrando que es necesario continuar el trabajo de desarrollo de nuevos métodos para atender casos similares.

Índice general

| | |
|--|-----------|
| Glosario | 7 |
| 1. Introducción | 9 |
| 1.1. Motivaciones en el cálculo del volumen de politopos | 9 |
| 1.2. Organización del documento | 10 |
| 2. Marco Teórico | 11 |
| 2.1. ¿Qué es un politopo? | 11 |
| 2.2. Cálculo de volumen con métodos exactos | 12 |
| 2.3. Cálculo de volumen con métodos aproximados | 13 |
| 2.3.1. Monte Carlo Directo | 13 |
| 2.3.2. Monte Carlo Multifases | 13 |
| 2.3.3. Evolución Monte Carlo Multifases | 14 |
| 2.3.4. Implementaciones estado del arte | 15 |
| 2.3.5. Métodos implementados en este trabajo | 16 |
| 2.3.6. Caminatas aleatorias | 16 |
| 2.4. Alcance del proyecto | 17 |
| 3. Redes metabólicas | 19 |
| 3.1. Modelos metabólicos | 19 |
| 3.2. Modelos estequiométricos | 20 |
| 3.3. Modelos estequiométricos estudiados | 21 |
| 3.3.1. Modelo ramificado simple | 21 |
| 3.3.2. Modelo núcleo de E. coli | 22 |
| 3.3.3. Modelo compacto E. coli | 22 |
| 4. Algoritmos implementados | 27 |
| 4.1. Calculo de Volumen sorteando muestras dentro del politopo | 27 |
| 4.2. Algoritmo Fok-Crevier | 28 |
| 4.3. Metodo basado en MCMC | 29 |
| 4.3.1. Metropolis-Hastings | 29 |
| 4.3.2. Estimacion del volumen utilizando integración termodinámica | 30 |
| 4.4. Punto inicial | 32 |
| 4.5. Restricciones con igualdades | 33 |
| 4.6. Detalles de la implementación | 34 |
| 5. Experimentación numérica | 37 |
| 5.1. Resultados VolEsti | 37 |
| 5.1.1. Análisis de resultados | 41 |
| 5.2. Resultados sorteando muestras dentro del politopo | 42 |
| 5.2.1. Análisis de resultados | 44 |
| 5.3. Resultados FC | 47 |
| 5.3.1. Análisis de resultados | 48 |
| 5.4. Resultados MCMC | 50 |
| 5.4.1. Análisis de resultados | 51 |
| 5.4.2. Análisis de los resultados obtenidos | 53 |
| 5.5. Resultados Modelo compacto E. coli | 54 |
| 5.5.1. Comparando volumen exacto con aproximaciones | 55 |
| 5.5.2. Ajuste del modelo metabólico. | 56 |
| 5.5.3. Análisis de resultados | 59 |

| | |
|---|-----------|
| 5.6. Resultados Modelo núcleo E. coli | 61 |
| 5.6.1. Análisis de resultados | 62 |
| 6. Conclusiones | 65 |
| 7. Bibliografía | 67 |
| A. Algoritmos estado del arte | 69 |
| A.1. Sequence of Balls (SoB) | 69 |
| A.2. Cooling Gaussians (CG) | 70 |
| A.3. Cooling Bodies (CB) | 76 |
| B. Caminatas Aleatorias | 83 |
| B.1. Random-Directions Hit-and-Run: | 83 |
| B.2. Coordinate-Directions Hit and Run: | 84 |
| B.3. Ball Walk: | 84 |
| B.4. Billiard Walk: | 84 |
| C. Modelos básicos | 85 |
| D. Ejemplo HR aplicado al modelo simple: | 87 |
| E. Ejemplo restricciones con igualdades | 89 |
| F. Ejemplo Metropolis-Hastings | 91 |
| G. Algoritmos implementados | 95 |
| G.1. HR | 95 |
| G.2. FC | 95 |
| G.2.1. Centro Fijo | 95 |
| G.2.2. Centro Variable | 96 |
| G.3. MCMC | 96 |
| G.4. Resultados computacionales | 97 |
| G.4.1. Resultados Modelo compacto E. coli CG y CB | 97 |
| G.4.2. Resultados Modelo núcleo E. coli CG y CB | 100 |

Glosario

B(0,1) hiper-esfera de centro cero y radio uno. En general $B(c, r)$ hiper-esfera de centro c y radio r . 13

BaW Ball Walk. 16

BiW Billiard Walk. 16

CB Cooling Balls. 15

CDHR Coordinate-Directions Hit and Run . 16

CG Cooling Gaussian. 15

FBA Flux Balance Analysis (Análisis de Balance de Flujos). 19

FC Fok Crevier. 27

GLPK GNU Linear Programming Kit. 34

HR Hit-and-Run. 83

LB Lower bound (límite inferior). 20

MCMC Markov Chain Monte Carlo. 16

MH Metropolis Hastings. 27

MMC Multiphase Monte Carlo. 13, 15

PT Parallel Tempering. 27, 32

RDHR Random-Directions Hit and Run. 16

S Matriz estequiométrica. 9

SA Simulated Annealing. 15

SoB Sequence of Balls. 6, 15, 69, 71, 72

UB Upper bound (límite superior). 20

W Base ortonormal del espacio nulo de la matriz estequiométrica. 28

Capítulo 1

Introducción

1.1. Motivaciones en el cálculo del volumen de politopos

El presente proyecto surge a partir del trabajo realizado en [34]. En el mismo se explora la aplicación de métodos de Monte Carlo en el estudio de redes metabólicas. Para esto, se plantea utilizar los modelos basados en restricciones, que son una representación matemática de todas las reacciones metabólicas que ocurren en un organismo. Estos modelos se basan en la utilización de una matriz estequiométrica, que tiene como entradas los coeficientes estequiométricos de los metabolitos que participan en una determinada reacción. En química, la estequiometría es el cálculo de las relaciones cuantitativas entre los reactivos y productos en el transcurso de una reacción química. Dichas reacciones son expresadas mediante ecuaciones químicas, y los coeficientes estequiométricos son los coeficientes de estas ecuaciones. Estos coeficientes representan el número de moléculas de un determinado tipo que participan en la reacción. Cada fila de la matriz estequiométrica representa un único metabolito, es decir una de las moléculas que participan en la reacción, y cada columna representa una única reacción. Imponiendo un balance de masas en todos los metabolitos de la red, se genera un sistema de ecuaciones que actúa como restricciones del modelo. El balance de masas se define en términos del flujo a través de cada reacción y los coeficientes estequiométricos de la reacción, generando un sistema de ecuaciones diferenciales. Asumiendo que la red llega a una condición estable de estado estacionario, las ecuaciones diferenciales se pueden expresar en notación matricial de modo que el conjunto de restricciones que la componen define un espacio donde se encuentran todas las posibles distribuciones de flujos. Por otro lado, se plantea un límite superior y un límite inferior para cada flujo, que pueden surgir de limitaciones termodinámicas o de otras limitaciones de capacidad de flujo.

En la figura (1.1) se resume la construcción de este tipo de modelos.

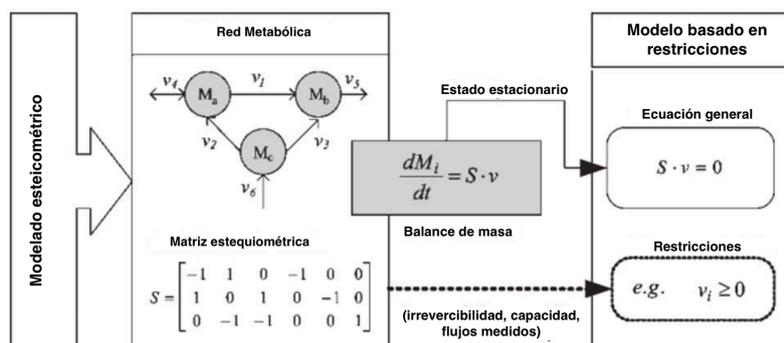


Figura 1.1: Modelos basados en restricciones, tomado de [34]

En donde los v_i representan a los flujos, los M_i representan a los metabolitos y S representa a la matriz estequiométrica.

Una parte importante del estudio de los modelos basados en restricciones, es el cálculo del volumen del espacio de soluciones. Determinar el tamaño y la forma de este espacio puede dar información acerca de los procesos metabólicos que ocurren dentro de una célula. Las restricciones estequiométricas restringen el espacio de posibles distribuciones de flujo a un hiperplano, subespacio de R^n . Además, si se incluyen las restricciones

de capacidad máxima de flujo, el espacio de soluciones en estado estacionario de una red metabólica representa un politopo convexo, representado en formato *H-politopo* [34].

En [34] se estudiaron tres modelos metabólicos diferentes denominados, modelo ramificado simple, modelo núcleo *Escherichia coli* (*E. coli*) y modelo compacto *Escherichia coli*.

El modelo ramificado simple está incluido en R^3 y se estudió debido a su simplicidad que permite visualizar y entender fácilmente los conceptos aplicados. Este modelo se utilizó para ilustrar el concepto del espacio de soluciones de un modelo metabólico y la relación de su dimensionalidad con la cantidad de reacciones y metabolitos del sistema.

El modelo núcleo de *E. coli* [29] es un modelo basado en restricciones simplificado, aunque contiene suficiente información del metabolismo de la bacteria como para llevar a cabo cálculos significativos. A partir de este modelo, en [34] se construye un modelo simplificado, que fue denominado modelo compacto *E. coli*, con menos dimensiones que permitió aplicar los métodos implementados en dicho trabajo.

Debido a que los métodos estudiados en [34] no permitieron realizar el cálculo del volumen del espacio de soluciones del modelo compacto en su totalidad y del modelo núcleo, es que el objetivo de este proyecto es el desarrollo e implementación de métodos de cálculo de volúmenes de politopos en altas dimensiones, tanto de manera exacta (en la medida que la dimensión del problema lo permita) como por aproximaciones basadas en algoritmos de Monte Carlo. Luego de estudiar dichos métodos, los aplicaremos a este caso práctico repitiendo algunos de los experimentos desarrollados en [34]. Como estos experimentos fueron realizados solo en el modelo compacto y reduciendo las dimensiones del mismo, en este trabajo nos proponemos extenderlos al modelo compacto en su totalidad, sin reducción de dimensión y también intentaremos aplicar los mismos experimentos en el modelo núcleo *E. coli*.

1.2. Organización del documento

Este documento se organiza de la siguiente manera: En el Capítulo 2 se presenta el marco teórico y el estado del arte en este campo. En el Capítulo 3, se presenta una breve descripción del problema biológico que motiva el trabajo. En el Capítulo 4 se introducen los conceptos que permiten entender los algoritmos de aproximación de volumen implementados. En el Capítulo 5 se presentan detalles de la implementación realizada y en el Capítulo 6 los resultados obtenidos al aplicar los algoritmos implementados y los del estado del arte. Por último, el Capítulo 7 presenta las conclusiones. Finalmente en los anexos se puede encontrar una descripción más en detalle de los algoritmos que se utilizaron en este trabajo, cuyas implementaciones se encuentran en bibliotecas open source. También se incluye el pseudocódigo de los algoritmos que fueron implementados y una serie de ejemplos que fueron estudiados a lo largo de este trabajo con el fin de entender los conceptos necesarios para las implementaciones realizadas.

Capítulo 2

Marco Teórico

Este capítulo presenta un marco teórico para el trabajo, en donde se intenta brindar un concepto general sobre lo que es un politopo, qué métodos existen para calcular su volumen de manera exacta y que métodos permiten aproximar el volumen cuando se trata de politopos que se encuentran en espacios de altas dimensiones. En la primera sección damos una breve descripción matemática de los politopos, para entender cómo es posible representarlos e identificar qué tipo de representación es la que se corresponde con las redes metabólicas. En la segunda y tercera sección se presenta un estado del arte sobre las formas de calcular o aproximar el volumen de un politopo respectivamente. Finalmente, en la cuarta sección, presentamos los objetivos y alcances de este proyecto.

2.1. ¿Qué es un politopo?

Primero, vamos a introducir una serie de conceptos que nos permitirán entender la definición de politopo y sus diferentes representaciones. Estos conceptos fueron estudiados en [31], [19] y [2]

Definición 2.1.1. Un conjunto $K \in R^n$ es **convexo** si para todo $x, y \in K$ la línea que conecta x, y queda completamente contenida en K , esto quiere decir que $:\lambda x + (1 - \lambda)y \in K, \forall x, y \in K, 0 \leq \lambda \leq 1$

Definición 2.1.2. Sea $K \subset R^n$ un conjunto cualquiera. Definimos la **envoltura convexa** de K , que se representa por $\text{conv}(K)$, como la intersección de todos los subconjuntos convexos de R^n que contienen a K . Se puede pensar en la envoltura convexa, como el menor conjunto convexo que contiene a K . Además, se puede asegurar que un conjunto $K \subset R^n$ será convexo si, y sólo si, la envoltura convexa de K es el propio K , esto es, $\text{conv}(K) = K$.

Definición 2.1.3. Se define la envoltura afín de K , y se representa por $\text{aff}(K)$, como la intersección de todos los subespacios afines que contienen a K . Se puede ver a $\text{aff}(K)$ como el menor subespacio afín que contiene a K .

$$\text{aff}(K) = \left\{ \sum_{i=1}^p \lambda_i x_i \mid x_1, \dots, x_p \in K, \sum_{i=1}^p \lambda_i = 1 \right\}$$

Definición 2.1.4. Si K es convexo, la **dimensión de K** es igual a la dimensión del subespacio afín más pequeño que contiene K . Entonces $\dim(K) = \dim(\text{aff}(K))$

Definición 2.1.5. Un **hiperplano H** de R^n se puede expresar como $H = \{x \in R^n : \langle x, u \rangle = c\}$, donde u es un vector unitario que es normal a H y c es una constante. De la misma manera, los dos **semiespacios cerrados que determina el hiperplano H** , H^+ y H^- , se representan como: $H^+ = \{x \in R^n : \langle x, u \rangle \geq c\}$ y $H^- = \{x \in R^n : \langle x, u \rangle \leq c\}$.

Definición 2.1.6. Dado un conjunto convexo y cerrado K de R^n , se dice que un **hiperplano H soporta al conjunto K en un punto $x \in K$** si $x \in K \cap H$ y K está contenido en H^+ o H^- .

Definición 2.1.7. Sea $K \in R^n$ un conjunto convexo y cerrado. Diremos que H es un **hiperplano soporte a K** si H soporta a K en algún punto x de dicho conjunto, el cual será, necesariamente, un punto de la frontera. Así mismo, si H es un hiperplano soporte a K y $K \cap H^-$, entonces diremos que H^- es un **semiespacio soporte a K** (análogamente para H^+).

Definición 2.1.8. V -politopo: Es la envoltura convexa de un número finito de puntos $X = \{x_1, \dots, x_d\}$ en R^n .

$$P = \text{conv}(X) := \left\{ \sum_{i=1}^d \lambda_i x_i \mid \lambda_i \geq 0, \sum_{i=1}^d \lambda_i = 1 \right\}$$

Definición 2.1.9. ***H-politopo:** Es la intersección de un número finito de semiespacios cerrados. Por lo que podemos verlo como el espacio de soluciones de un sistema finito de desigualdades lineales.*

$$P = P(A, b) := \{x \in \mathbb{R}^n \mid a_i^T x \leq b_i, 1 \leq i \leq m\}$$

Donde $A \in \mathbb{R}^{m \times n}$ es una matriz con filas a_i^T , y $b \in \mathbb{R}^m$ es un vector con entradas b_i .

Definición 2.1.10. *Un **Politopo** es un conjunto $P \in \mathbb{R}^n$ que puede ser representado como un V -politopo y como un H -politopo. El teorema principal de la teoría de politopos establece que las definiciones de V -politopos y de H -politopos son equivalentes. Es decir, cada V -politopo tiene una descripción mediante un sistema acotado de desigualdades, y cada H -politopo puede obtenerse como la envoltura convexa de un conjunto finito de puntos (sus vértices).*

Debido a que la representación de los modelos metabólicos definen un H -politopo, en este trabajo nos centraremos en los algoritmos de cálculo y aproximación de volumen que hagan uso de esa representación.

2.2. Cálculo de volumen con métodos exactos

Se sabe que el problema de calcular el volumen de los politopos (dados en términos de sus vértices, V -politopos, o en términos de sus restricciones, H -politopos) es $\#P$ -hard [9]. Existen varios algoritmos exactos [13], que permiten calcular el volumen, sin embargo, dichos algoritmos, no pueden manejar politopos generales para dimensión $d > 15$.

En este trabajo investigamos cómo usar dos softwares diferentes (Vinci [3] y Latte[24]) que implementan dichos algoritmos, y que utilizamos para poder validar los resultados de obtenidos con los métodos de aproximación de volumen en una serie de politopos que fueron utilizados para testear estos softwares.

En [13] explican que todos los métodos de cálculo de volumen determinista descomponen un politopo dado en politopos cuyos volúmenes son más fáciles de calcular. El volumen de los politopos simples (politopos de dimensión d que tienen $d + 1$ vértices) se puede obtener a partir del cálculo de un determinante¹, por lo que estos algoritmos hacen la descomposición del politopo original en simples para poder calcular su volumen en función de los volúmenes de la descomposición. Dependiendo del tipo de descomposición en simples que se haga, existen dos clases básicas de algoritmos, Triangulaciones y Descomposiciones con signo.

- **Triangulaciones:** Una triangulación de un politopo P es una colección finita $\Delta_i : i = 1, \dots, s$ de simples tal que $P = \bigcup_{i=1}^s \Delta_i$ y la intersección de cualquier par de simples es la cara que tienen en común. Esta es la definición habitual de una triangulación, pero para propósitos de cálculo de volumen sería suficiente exigir que no haya dos simples que compartan un punto interior. Entonces $Vol(P) = \sum_{i=1}^s Vol(\Delta_i)$
- **Descomposiciones con signo:** La restricción de que no hay dos simples que puedan compartir un punto interior no es necesaria. Si no se cumple, algunas partes del politopo están cubiertas más de una vez, lo que debe corregirse restando sus volúmenes respectivos. Una descomposición con signo de P es, una colección finita $\{(\Delta_i, \sigma_i) : i = 1, \dots, s\}$ de simples Δ_i y signos $\sigma_i \in \{\pm 1\}$ con la siguiente propiedad: Si un punto de P no se encuentra en el límite de ningún simple, entonces el número de simples positivos que lo contienen excede el número de simples negativos que lo contienen en 1. Un punto fuera de P está contenido en tantos simples positivos como negativos. Resulta que: $Vol(P) = \sum_{i=1}^s \sigma_i Vol(\Delta_i)$

El software Vinci implementa las descomposiciones Delaunay triangulation, Boundary Triangulation, Triangulation by Cohen and Hickey, Lawrence's Signed Decomposition, Lasserre's Signed Decomposition y Hybrid orthonormalization technique, que estan basadas en las triangulaciones y las descomposiciones con signo.

El software LattE puede calcular integrales de polinomios y volúmenes de politopos. En [8], describen brevemente los conceptos matemáticos que se utilizan para la implementación de este software y plantean que para calcular las integrales de polinomios es necesario contar con una descomposición geométrica de los politopos sobre los cuales se integra en esas piezas. Un caso especial de la integración es el cálculo del volumen de un politopo que es equivalente a integrar el monomio 1 sobre el politopo. Esto se resuelve en la implementación utilizando las mismas técnicas que para la integración (triangulaciones o descomposiciones con signo).

Las implementaciones de Vinci² y LattE³ están hechas en C++. El código está disponible gratuitamente bajo la licencia GNU.

¹<https://es.wikipedia.org/wiki/S%C3%ADmplex>

²El manual de Vinci puede consultarse en <https://www.math.u-bordeaux.fr/~aenge/software/vinci/manual.pdf>

³El manual de Latte puede consultarse en https://www.math.ucdavis.edu/~latte/software/packages/latte_current/manual_v1.7.2.pdf

Es importante destacar que los métodos de cálculo de volumen, implementados por ambos softwares, solo funcionan con politopos de dimensión completa. Un politopo es de dimensión completa si es un objeto de dimensión n en R^n .

2.3. Cálculo de volumen con métodos aproximados

Esta sección trata sobre diferentes métodos basados en Monte Carlo que permiten aproximar el volumen de un Politopo. Primero, presentamos una descripción de la forma en la cual se aplica Monte Carlo directo para estimar el volumen de un politopo y las dificultades que presenta aplicar este método en espacios de altas dimensiones. Luego, en las secciones 2.3.2 y 2.3.3, presentamos el marco teórico del método Monte Carlo Multifase y su evolución presentada en diferentes trabajos a lo largo de las últimas décadas. En la sección en la sección 2.3.4 presentamos las implementaciones basadas en el enfoque MMC que encontramos implementadas en librerías open source. Luego, en la sección 2.3.5, comentamos los métodos que encontramos en la literatura relacionada directamente con la aproximación de volumen para modelos metabólicos y que fueron implementados en este trabajo. Finalmente, en la sección 2.3.6, presentamos una breve descripción de un punto clave en el cual están basados todos los métodos estudiados que son las caminatas aleatorias.

2.3.1. Monte Carlo Directo

Para aproximar el volumen de un politopo P aplicando el método de Monte Carlo directo, lo que se hace es encontrar un espacio de referencia en el que se pueda calcular su volumen y que contenga a P (lo mejor es que sea lo más ajustado posible, pero podría ser R^n). Dentro de este espacio de referencia se sortean puntos aleatorios. Y luego, considerando la proporción de puntos sorteados que pertenecen P , se puede aproximar el volumen del politopo. Es decir que, siendo R el espacio de referencia, K el número total de muestras (distribuidas uniformemente en R) y M la cantidad de muestras sorteadas que pertenecen a P , el volumen se aproxima como:

$$vol(P) \approx vol(R) \frac{M}{K}$$

Aplicar el método de Monte Carlo directo en espacios de altas dimensiones para estimar el volumen de un politopo es ineficiente, debido a que la diferencia entre el volumen del espacio de referencia y el volumen buscado crece a medida que crece la dimensión. Esto hace que la probabilidad de sortear un punto que caiga dentro del politopo se haga más pequeña.

En [32] se presenta un algoritmo capaz de aproximar el volumen de un politopo en tiempo polinomial (Monte Carlo Multifase, MMC por sus siglas en ingles, Multiphase Monte Carlo). Al principio la complejidad teórica de este algoritmo era de $O(n^{23})$, y a lo largo de las últimas décadas se fue reduciendo esta complejidad hasta llegar a $O(n^3)$ [7].

2.3.2. Monte Carlo Multifases

Como se explica en [32] el método de Monte Carlo Multifases aborda esta dificultad dividiendo el problema. Para esto se expresa el volumen de P en función de la relación entre los volúmenes de una serie de cuerpos $K_i \in R^n$ con $i = 0 \dots m := \lceil 4n \log n \rceil$ y $K_{i-1} \subseteq K_i$. Estos cuerpos están uno incluido dentro del otro, lo que hace que se pueda estimar la relación entre el volumen de dos de ellos aplicando Monte Carlo. Entonces el volumen de P queda expresado como un producto $V = \prod v_i$ donde cada v_i puede ser aproximado por métodos de Monte Carlo. La cantidad de v_i no puede ser muy grande y además estos deben estar acotados.

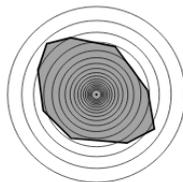


Figura 2.1: Monte Carlo Multifase, tomado de [32]

Para obtener los valores v_i , se considera un $P \in R^n$ que contiene a la hiper-esfera de centro cero y radio uno $B(0, 1)$, y que al mismo tiempo está contenido en la hiper-esfera de centro cero y radio $(n + 1)\sqrt{n}$, $B(0, (n + 1)\sqrt{n})$. En [32] mencionan que se puede calcular el diámetro de esta hiper-esfera en orden $O(n^4)$.

Se denomina sandwiching al proceso de encontrar los elipsoides inscrito y circunscrito, que son necesarios para determinar los K_i .

Entonces a partir de la siguiente secuencia de intersecciones de P con las hiper-esferas de centro cero y radio $2^{i/n}$:

$$K_i = P \cap B(0, 2^{i/n}) \text{ con } i = 0 \dots m := \lceil 4n \log n \rceil$$

Donde, $K_0 = B(0, 1)$ y $K_m = P$.

Se define $v_i = \frac{\text{vol}(K_i)}{\text{vol}(K_{i-1})} \forall i = 1 \dots m$.

Dado que $K_{i-1} \subseteq K_i$ y $K_i \subseteq \frac{2^{i/n}}{2^{(i-1)/n}} K_{i-1} = 2^{1/n} K_{i-1}$ se cumple que $1 \leq v_i \leq 2$

Por lo tanto, generando puntos distribuidos uniformemente en cada cuerpo convexo K_i se puede aproximar $\text{vol}(K)$ como:

$$\text{vol}(P) = \text{vol}(K_0) \frac{\text{vol}(K_1)}{\text{vol}(K_0)} \frac{\text{vol}(K_2)}{\text{vol}(K_1)} \dots \frac{\text{vol}(K_m)}{\text{vol}(K_{m-1})} = \text{vol}(K_0) \prod_{i=1}^m \frac{\text{vol}(K_i)}{\text{vol}(K_{i-1})}$$

Donde $\text{vol}(K_0) = \text{vol}(B(0, 1)) = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$

Esto reduce el problema a estimar las relaciones $\text{vol}(K_i)/\text{vol}(K_{i-1})$, y este problema se resuelve generando N puntos distribuidos uniformemente en K_i y contando cuántos de ellos caen en K_{i-1} .

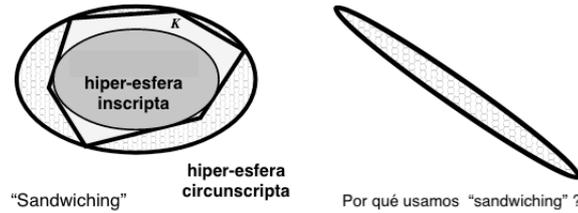


Figura 2.2: Importancia del redondeo, tomado de [32]. Al proceso de encontrar una hiper-esferas que contenga al politopo y otra que esté incluida en él, se le denomina “sandwiching”

Sin embargo, este tipo de algoritmos no funcionan bien para cuerpos con formas alargadas (figura 2.2), es decir, cuerpos que son demasiado largos en una dirección en comparación con sus otras dimensiones. Porque si bien el problema de las proporciones se vería atenuado, seguiría existiendo una diferencia considerable entre el politopo y la hiper-esfera que lo contiene y también por la dificultad de sortear puntos aleatorios dentro de un cuerpo alargado. Por lo que antes de aplicarlos es importante aplicar al politopo un proceso de redondeo. El redondeo significa que se trata de encontrar una transformación lineal A tal que la relación de los radios de las bolas circunscritas e inscritas al politopo AP , (que es el resultado de aplicar la transformación lineal A a P) es pequeña. La relación de redondeo influye mucho en la velocidad de estos algoritmos.

Para generar puntos uniformemente distribuidos en cada K_i se utilizan recorridos aleatorios y se hace de forma recursiva, aprovechando que en cada etapa del algoritmo se cuenta con un v_i casi uniformemente distribuido en K_{i-1} , como $K_{i-1} \in K_i$ se puede iniciar otro recorrido aleatorio en K_i partir de este.

2.3.3. Evolución Monte Carlo Multifases

Variando los métodos que se utilizan para el redondeo, para el sandwiching y la forma en que se generan los puntos distribuidos uniformemente dentro de cada K_i , en [15] y [11] se implementan diferentes versiones de este enfoque de Monte Carlo Multifase.

En [12], [6], [7], se mejoraron los resultados del enfoque MMC, teniendo en cuenta que el cálculo del volumen es un caso especial de integración. Entonces en lugar de una secuencia de cuerpos, para expresar el volumen como $V = \prod v_i$ se construye una secuencia de funciones $\{f_0 \leq \dots \leq f_{m-1}\}$ $f_i : R^d \rightarrow R$ que “conectan” una función f_0 a la función característica f_m de P .

$$f_m(x) = \begin{cases} 1, \forall x \in P \\ 0, \forall x \notin P \end{cases}$$

La integral $\int_P f_0(x) dx$ debe ser fácil de calcular.

$$\text{vol}(P) = \int_P 1 dx = \int_P f_0(x) dx \frac{\int_P f_1(x) dx}{\int_P f_0(x) dx} \frac{\int_P f_2(x) dx}{\int_P f_1(x) dx} \cdots \frac{\int_P 1 dx}{\int_P f_{m-1}(x) dx}$$

Si las f_i son funciones características del K_i , entonces este es solo el algoritmo estándar. La ganancia proviene del hecho de que el número de puntos de muestra necesarios en cada fase es menor si los f_i son suaves (funciones que admiten derivadas de cualquier orden continuas).

Este método, puede verse como una variación del algoritmo de enfriamiento simulado SA [23] (por sus siglas en inglés Simulated annealing). Es un método de búsqueda aleatoria de propósito general para la optimización. Este método realiza un muestreo aleatorio en el espacio de posibles soluciones, ajustando gradualmente un parámetro llamado “temperatura”. A alta temperatura, la caminata aleatoria converge rápidamente a la distribución uniforme en todo el espacio. A medida que desciende la temperatura, la distribución estacionaria se inclina cada vez más hacia las soluciones óptimas.

Finalmente, en [4] se propone una nueva estimación de volumen práctica para politopos convexos, mejorando los métodos anteriores para H -politopos, al tiempo que proporciona el primer método capaz de escalar en el caso de V -politopos. En este nuevo método, se utiliza una secuencia adaptativa de cuerpos convexos, enfriamiento simulado y propiedades estadísticas de las proporciones que definen el volumen para reducir el número de fases en MMC, así como el tamaño de muestra requerido para estimar estas proporciones. Se aprovecha el hecho de que el muestreo de rechazo funciona de manera eficiente para proporciones acotadas. Para reducir el número de fases, se emplean pruebas estadísticas para encontrar los ratios $r_i = \text{vol}(P_i + 1)/\text{vol}(P_i)$ con probabilidades elevadas. El nuevo programa de enfriamiento no necesita un cuerpo que contenga a P , basta con establecer $P_0 = P$. Además, este método no requiere calcular una hiper-esfera inscrita, como hacen los métodos anteriores, la hiper-esfera con el volumen mínimo se calcula al mismo tiempo que se establece el programa de enfriamiento. También se aprovecha el hecho de que cuando se muestrean N puntos uniformes de P_i , el número de puntos en P_{i+1} sigue la distribución binomial. Por lo que se aprovecha el intervalo de confianza de la proporción binomial y se modifica utilizando la desviación estándar de una ventana deslizante, para especificar un nuevo criterio de convergencia empírico para la razón, con el fin de minimizar la cantidad de muestras N .

2.3.4. Implementaciones estado del arte

Según [5] actualmente, en la práctica, el límite en la dimensión para la aproximación del volumen es de unos pocos cientos. El primer algoritmo práctico que escaló a grandes dimensiones fue el de [12] (Sequence of Balls o algoritmo SoB), su implementación fue hecha en C++. Luego, en [6] se presentó la implementación de otro algoritmo asintóticamente más rápido (Cooling Gaussians o algoritmo CG), implementado en MATLAB. Ambos algoritmos pueden manejar solo politopos en su representación H -politopo en dimensiones altas. En [4] se presenta un algoritmo (Cooling Bodies o algoritmo CB) que está diseñado para escalar en grandes dimensiones para otras representaciones de politopos.

SoB se basa en el enfoque MMC, utilizando la secuencia de cuerpos para generar los v_i , CG se basa en la construcción de una secuencia de funciones que conectan una función gaussiana a la función característica de P . Por otra parte CB es la nueva propuesta que utiliza una secuencia adaptativa de cuerpos convexos, enfriamiento simulado y propiedades estadísticas de las proporciones que definen el volumen reduciendo el número de fases en MMC.

En el anexo A se puede encontrar una descripción más detallada de estos tres métodos.

La implementación de estos tres algoritmos podemos encontrarla en la librería implementada en C++ llamada **VolEsti**. El código fuente de esta librería puede encontrarse en https://github.com/GeomScale/volume_approximation bajo la licencia GNU.

| Algoritmo | H -Politopo | | V -Politopo |
|-------------------|----------------------|-------------------|---------------|
| | $dimension \leq 200$ | $dimension > 200$ | |
| SoB (2014) | | | |
| CG (2016) | | ✓ | |
| CB (2019) | ✓ | | ✓ |

Tabla 2.1: Performance de los algoritmos implementados en VolEsti. Los algoritmos que son más apropiados para los diferentes escenarios son los que están marcados en la tabla. Datos tomados de [5].

En la tabla 2.1 se resume el desempeño de los tres métodos prácticos implementados en VolEsti que proporcionan en [5]. Allí plantean que el algoritmo CB es la opción más eficiente para H -politopos en menos de 200 dimensiones y para V -politopos en cualquier dimensión. Para el resto de los casos, se debe elegir el algoritmo CG.

Las pruebas preliminares muestran que el algoritmo SoB tiene la mayor probabilidad de aproximar el volumen (P) dentro de un error relativo objetivo, pero no está claro si esta es la regla.

2.3.5. Métodos implementados en este trabajo

Por otra parte, en [21] se presentan dos estimadores diferentes, el estimador Fok-Cremer (FC) [14] y un estimador basado en Monte Carlo Markov Chains (MCMC) aplicando la técnica de integración termodinámica. Según [11] este enfoque ofrece solo garantías de convergencia experimentales, a diferencia de los anteriores cuyas garantías teóricas han sido ampliamente estudiadas. En [26] se aplican estos dos enfoques para estimar el volumen del modelo *Escherichia coli* (*E. coli*) de tamaño mediano. Los resultados que obtienen demuestran que el estimador MCMC converge en tiempo polinomial mientras que para el estimador FC el tiempo de cálculo crece exponencialmente con las dimensiones del problema, ambos estimadores convergen asintóticamente al volumen real del politopo. En este trabajo realizamos una implementación en python de estos algoritmos.

2.3.6. Caminatas aleatorias

Una parte importante de todos estos algoritmos, es la forma que que se generan N puntos uniformemente distribuidos, dentro de P . Para esto, se utilizan diferentes tipos de caminatas aleatorias, que son una ruta definida como una serie de pasos aleatorios.

1. Random-Directions Hit and Run (RDHR)
2. Coordinate-Directions Hit and Run (CDHR)
3. Ball Walk (BaW)
4. Billiard Walk (BiW)

Todas estas caminatas, excepto BiW, se pueden usar para muestrear aproximadamente de cualquier distribución de probabilidad truncada en P (se dice que una distribución de probabilidad para una variable aleatoria X se trunca cuando se excluye algún conjunto de valores en el rango de X .), mientras que BiW solo se puede usar para generar puntos aproximadamente distribuidos uniformemente en P .

En el anexo B damos una descripción breve que encontramos en [5] de las caminatas aleatorias que están implementadas en VolEsti.

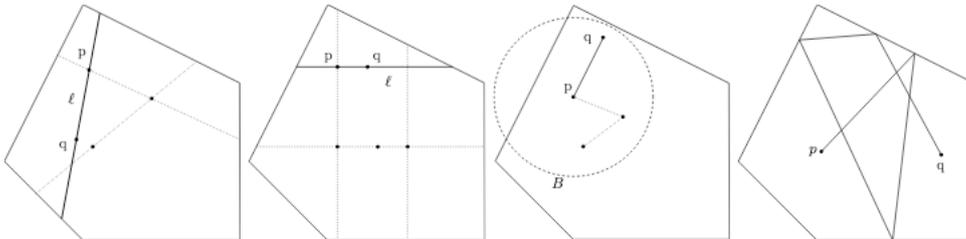


Figura 2.3: Imagen tomada de [5], ejemplos caminatas aleatorias. De izquierda a derecha: RDHR, CDHR, BaW, BiW. p es el punto en el paso actual y q es el nuevo punto sorteado

En general, al iniciar una caminata aleatoria desde un punto $p \in P$, cuantos más puntos se generan, menos correlacionados estarán estos puntos. El número de pasos que hay que dar para obtener un punto independiente de los anteriores, se llama tiempo de mezcla (mixing time). El costo por cada paso es el número de operaciones realizadas para generar un punto y el costo total para generar un punto aleatorio es el tiempo de mezcla por el costo de cada paso.

| <i>Random Walk</i> | <i>Tiempo de mezcla</i> | Costo por paso H-politopo | <i>Costo por paso</i> V-politopo |
|--------------------|-------------------------|------------------------------|-------------------------------------|
| RDHR | $O(d^3)$ | $O(md)$ | 2 PLs |
| CDHR | ? | $O(nd)$ | 2 PLs |
| BaW | $O(d^{2,5})$ | $O(md)$ | 1 PL |
| BiW | ? | $O((d + R)m)$ | R PLs |

Tabla 2.2: Resumen de las caminatas aleatorias implementadas en **VolEsti**. Donde PL refiere a un problema de programación lineal y R a el número de rechazos por punto en BiW. d es la dimensión y m la cantidad de restricciones que componen al politopo en formato *H-politopo*. Información tomada de [5]

En el cuadro 2.2, tomado de [5], se puede ver un resumen de los valores conocidos para los tiempos de mezcla y costo por paso de estas caminatas aleatorias. Para el tiempo de mezcla de RDHR se asume que P está bien redondeado. Para el tiempo de mezcla de BaW se asume que P está en posición isotrópica. No se conocen límites teóricos sobre el tiempo de mezcla para CDHR y BiW.

2.4. Alcance del proyecto

La motivación de este trabajo es estudiar la aplicabilidad de métodos de Monte Carlo para estimar el volumen del espacio de soluciones determinado por las posibles configuraciones de las redes metabólicas de diferentes microorganismos, como por ejemplo *Escherichia coli*. Esta motivación surge a partir del trabajo previo realizado en [34]. En dicho trabajo, se realizó un estudio sobre un modelo simplificado de *E. coli* llamado modelo compacto. Las técnicas implementadas en dicho trabajo no permitieron calcular el volumen del modelo compacto en su totalidad. En su lugar, realizaron experimentos apagando y prendiendo diferentes reacciones con el fin de reducir la dimensión del modelo. De esta manera se pudo realizar el análisis sobre el comportamiento del microorganismo en las condiciones biológicas representadas por la ausencia de dichas reacciones. Es por este motivo que el objetivo principal de este trabajo es estudiar la posibilidad de realizar la estimación de volumen aplicando métodos de Monte Carlo en los modelos compacto y núcleo de *E. coli* en su totalidad, con una precisión razonable en tiempos de cálculo limitados. Es decir considerando todas las reacciones que involucran a cada modelo. Por otra parte, nos planteamos la necesidad de contar con por lo menos dos métodos que nos permitan aproximar el volumen de dichos modelos. De esta manera podremos realizar una comparación entre ambos resultados con la cual podríamos llegar a descartar malos resultados.

Para esto, primero vamos a implementar la técnica que se basa en la utilización de muestreos utilizando Hit and Run desarrollada en [34] y estudiar el comportamiento aplicando las caminatas aleatorias implementadas en la librería VolEsti, RDHR, CDHR, BaW, BiW. Luego vamos a estudiar e implementar los métodos FC y MCMC que fueron utilizados en [26] para aproximar el volumen de *E. coli*. Para validar los métodos implementados, y los que vienen incluidos en la librería de VolEsti, vamos a realizar pruebas y estudiar el comportamiento de los cuatro algoritmos aplicados a un conjunto de politopos básicos, de los cuales conocemos su volumen exacto. De esta manera contaremos con cinco estimadores (FC, MCMC y SoB, CG y CB implementados en VolEsti), que nos permitan comparar resultados cuando las dimensiones crecen. Cuando sea posible, los resultados se van a comparar contra el volumen exacto. Por último, vamos a utilizar como casos de prueba algunos de los experimentos relacionados al cálculo del volumen que se realizaron en [34] y vamos a aplicar los nuevos métodos para aproximar el volumen de los modelos *E. coli* núcleo y compacto.

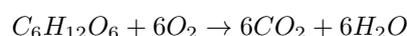
Capítulo 3

Redes metabólicas

En este capítulo se intenta brindar una idea general de lo que es una red metabólica. Se busca explicar que es un modelo basado en restricciones, como es posible conseguirlo y la importancia que tiene para su análisis los modelos estequiométricos y el estado estacionario en dichos modelos. Luego se presenta una breve descripción de los modelos estudiados en [34], en los cuales aplicaremos los métodos de aproximación de volumen mencionados en la sección anterior.

3.1. Modelos metabólicos

El conjunto de reacciones bioquímicas que se producen dentro de la célula representan el metabolismo [20]. Estas reacciones son catalizadas por enzimas y ocurren con el propósito de completar procesos de transporte, síntesis y degradación de moléculas que permiten a la célula obtener energía química del entorno y almacenarla, para luego usarla en diferentes funciones celulares. En su estructura interna, el metabolismo está compuesto por un conjunto de metabolitos interconectados a través de las reacciones bioquímicas, lo que da lugar a una compleja red de transformaciones. Un ejemplo de un proceso metabólico es la degradación de la glucosa. Muchas células obtienen energía de la glucosa ($C_6H_{12}O_6$) mediante un proceso llamado respiración celular. Durante este proceso, una molécula de glucosa se degrada gradualmente, en muchos pasos pequeños. Este proceso tienen la siguiente reacción general:



En los modelos metabólicos que se utilizan en este trabajo cada fila de la matriz estequiométrica representa una de las moléculas que participan en la reacción, y cada columna representa una única reacción.

Las redes metabólicas son modelos matemáticos que describen el metabolismo de la célula. Según [17] el desarrollo de técnicas rápidas y eficientes de secuenciación de ADN han permitido la obtención de reconstrucciones metabólicas a nivel de genoma de una manera sistemática para gran cantidad de organismos.

A partir del genoma de una especie y de los genomas de otras especies conocidas, se identifican los genes que codifican enzimas. Las enzimas son las que generan las reacciones que componen la red metabólica, entonces para cada enzima identificada se asume que cataliza las mismas reacciones que en otras especies.

Al finalizar este proceso se tiene un primer borrador del metabolismo de la especie. Pero, como este primer borrador puede estar incompleto, es decir que faltan reacciones químicas, es necesario aplicar métodos computacionales que permitan solucionar esta falta de información, y luego validar dichas reconstrucciones a través de experimentos para así obtener reconstrucciones metabólicas.

Aunque este tipo de reconstrucción metabólica no es perfecta, en general es lo suficientemente completa como para obtener un análisis válido, aplicando métodos como el análisis de balance de flujos (FBA, del inglés flux balance analysis) que resuelve un problema de programación lineal para encontrar el valor máximo de un flujo (por ejemplo, el flujo que modela la producción de biomasa) bajo ciertas condiciones.

Existen distintas bases de datos como EcoCyc¹ y BioCyc² donde se pueden encontrar mapas completos del metabolismo de diferentes especies que incluye diversas capas de información: genética, enzimática, molecular, etc. Por ejemplo, la cepa MG1655 de *Escherichia coli* (*E. coli*) K-12 tiene un genoma completamente secuenciado. El plano genético revela una visión de todos los aspectos metabólicos y catabólicos de este organismo modelo. También proporciona un paradigma para comprender otras bacterias [22]. *Escherichia coli* es un organismo modelo utilizado frecuentemente en el laboratorio por su velocidad de crecimiento, pocos requerimientos nutricionales y su amplia bibliografía.

¹<https://ecocyc.org>

²[urlhttps://biocyc.org](https://biocyc.org)

3.2. Modelos estequiométricos

Una vez que una reconstrucción metabólica se convierte en un modelo computacional, se puede utilizar para un gran número de aplicaciones. Dichos modelos se han utilizado para abordar un amplio espectro de aplicaciones básicas y prácticas en categorías como estudios de procesos evolutivos y análisis de propiedades de redes. En el análisis de balance de flujo (FBA), la reacción de la biomasa se puede optimizar resolviendo un problema de programación lineal para simular el crecimiento. El resultado es una predicción del flujo de estado estacionario a través de cada reacción en el modelo, incluida una predicción de la tasa de crecimiento equilibrada máxima de la célula. El crecimiento se puede simular en muchas condiciones diferentes, como condiciones aeróbicas o anaeróbicas, o crecimiento en glucosa u otros sustratos. Se simulan diferentes condiciones cambiando las restricciones en las reacciones de intercambio. Por ejemplo, las condiciones anaeróbicas se simulan cuando el límite inferior de la reacción de intercambio de O_2 se limita a un flujo cero, lo que no permite que entre O_2 en el sistema metabólico [29].

Para aplicar este tipo de análisis se construyen modelos estequiométricos a partir de la información de los modelos metabólicos. Se asume que la concentración de metabolitos intracelulares se encuentra en estado estacionario es decir, en un estado en que las concentraciones de metabolitos son constantes en el tiempo. Esto implica que la sumatoria de los flujos de las reacciones que producen un determinado metabolito debe ser igual a la sumatoria de los flujos de reacciones que consumen este mismo metabolito, es decir, que los flujos de producción y de consumo para todos los metabolitos estén balanceados.

Estos modelos se basan en la utilización de una matriz estequiométrica, que tiene como entradas los coeficientes estequiométricos de los metabolitos que participan en una determinada reacción. Estos coeficientes representan el número de moléculas de un determinado tipo que participan en la reacción. También se puede definir un límite superior e inferior para cada flujo. Estos límites pueden ser usados para incorporar datos experimentales, permitiendo restringir los flujos de consumo y producción de metabolitos a los valores determinados experimentalmente.

El sistema de ecuaciones e inecuaciones en conjunto con los límites superiores e inferiores para los flujos sirve para definir un espacio de soluciones, es decir, valores de flujo posibles. Para seleccionar las mejores soluciones dentro de este espacio se suele definir un objetivo biológicamente deseable en el sistema, por ejemplo la maximización de uno o más flujos. El objetivo biológico fijado dependerá del sistema estudiado.

Como se mencionó anteriormente, cada fila de la matriz estequiométrica representa un único metabolito, y cada columna representa una única reacción. Imponiendo un balance de masas en todos los metabolitos de la red, se genera un sistema de ecuaciones que actúa como restricciones del modelo. El balance de masas se define en términos del flujo a través de cada reacción y los coeficientes estequiométricos de la reacción, generando un sistema de ecuaciones diferenciales:

$$\frac{dC}{dt} = S \cdot V$$

donde C es el vector de concentraciones de metabolitos intracelulares, S es la matriz estequiométrica y V es el vector de flujos metabólicos. Asumiendo que la red está en estado estacionario, las ecuaciones diferenciales se pueden expresar en notación matricial de la siguiente forma:

$$S \cdot V = 0$$

Esta ecuación define un espacio donde se encuentran todas las posibles distribuciones de flujo V . Si S es de rango completo, hay m ecuaciones independientes, una por cada concentración de metabolito. Siendo el número de flujos n típicamente mayor que m , el sistema es indeterminado con $n - m$ grados de libertad.

Por otro lado, se plantea un límite superior y un límite inferior para cada flujo, que pueden surgir de limitaciones termodinámicas o de otras limitaciones de capacidad de flujo. Esto genera una ecuación para cada flujo de la forma:

$$v_j^{LB} \leq v_j \leq v_j^{UB}$$

donde v_j^{LB} y v_j^{UB} representan el límite inferior (LB) y superior (UB) de v respectivamente.

Determinar el tamaño y la forma de este espacio puede dar información acerca de los procesos metabólicos que ocurren dentro de una célula. Este volumen está relacionado con la capacidad de adaptación que tenga el organismo a cambios en las condiciones externas. Las restricciones estequiométricas restringen el espacio de posibles distribuciones de flujo a un hiperplano, subespacio de R_n . Además, si se incluyen las restricciones de capacidad máxima de flujo, el espacio de soluciones en estado estacionario de una red metabólica representa un politopo convexo, representado en formato $H - \text{politopo}$ [34].

3.3. Modelos estequiométricos estudiados

En esta sección se presentan los tres modelos que fueron estudiados en [34]. Dichos modelos serán utilizados más adelante para aplicar los algoritmos de aproximación de volumen estudiados.

3.3.1. Modelo ramificado simple

Este modelo es estudiado en [34] con el fin de facilitar la comprensión de los conceptos aplicados. Es un modelo para el cual se puede calcular fácilmente el volumen exacto de su espacio de soluciones. Además al tratarse de un modelo de dos dimensiones nos permite visualizar claramente los resultados obtenidos.

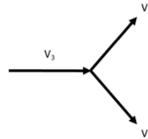


Figura 3.1: Modelo ramificado simple, tomado de [34]

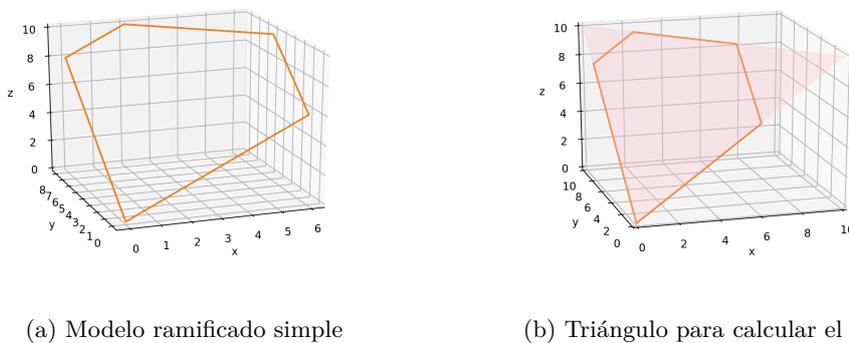
El modelo tiene un flujo de entrada V_3 y dos flujos de salida V_1 y V_2 , involucrando a un solo metabolito A . En la tabla (3.1) se puede ver la matriz estequiométrica del modelo. Las cotas superiores e inferiores para los tres flujos se pueden ver en la tabla 3.2

| | V_1 | V_2 | V_3 |
|-----|-------|-------|-------|
| A | -1 | -1 | 1 |

Tabla 3.1: Matriz estequiométrica del modelo ramificado simple. Tomado de [34]

| | LB | UB |
|-------|------|------|
| V_1 | 0 | 6 |
| V_2 | 0 | 8 |
| V_3 | 0 | 10 |

Tabla 3.2: Cotas inferiores (LB) y cotas superiores (UB) del modelo ramificado simple. Tomado de [34]



(a) Modelo ramificado simple

(b) Triángulo para calcular el área

Figura 3.2: Modelo ramificado simple

Como en este caso, el espacio de soluciones es un plano (figura 3.2a), podemos calcular su “volumen” (es decir el área) utilizando la siguiente fórmula:

Área de un triángulo de lados a , b y c : $\sqrt{s(s-a)(s-b)(s-c)}$ con $s = (a + b + c)/2$

Para esto calculamos el área del triángulo que contiene al espacio de soluciones (figura 3.2b) y luego le restamos el área de los dos triángulos que se forman al tener en cuenta las restricciones sobre el mismo.

$$Vol(P) = \sqrt{7500}\sqrt{192}\sqrt{12} = 69,28$$

3.3.2. Modelo núcleo de E. coli

El modelo núcleo de E. coli [29] es un modelo basado en restricciones simplificado, aunque contiene suficiente información del metabolismo de la bacteria como para llevar a cabo cálculos significativos. La representación matricial de este modelo consiste en una matriz estequiométrica de 72 filas (que corresponden a 72 metabolitos) por 95 columnas (que corresponden a 95 reacciones). Este modelo es un subconjunto de la reconstrucción metabólica a escala genómica iAF1260³ que contiene 1668 metabolitos y 2382 reacciones⁴. Este modelo se utiliza con fines educativos, ya que los resultados de la mayoría de los cálculos basados en restricciones son más fáciles de interpretar en esta escala más pequeña. También es útil para probar nuevos métodos de análisis basados en restricciones.

3.3.3. Modelo compacto E. coli

En [34] se toma el modelo núcleo como base para generar un modelo más simplificado para poder aplicar los métodos de Monte Carlo estudiados. El modelo creado fue denominado como “núcleo compacto de E. coli”. El sistema tiene un total de 17 metabolitos y 25 flujos. En la tabla 3.3 se puede ver la matriz estequiométrica del modelo y en la tabla 3.4 tenemos las restricciones de flujo. Antes de hacer cualquier análisis sobre estos datos es necesario eliminar las filas linealmente independientes de la matriz estequiométrica. Los metabolitos que corresponden a estas filas son: ATP, NADH, NADP y q8h2. Eliminando estos metabolitos se pasa a trabajar con una matriz de rango completo, que tiene un total de 25 flujos y 13 metabolitos, con una dimensión de su espacio de soluciones de 12.

³<https://systemsbiology.ucsd.edu/Downloads/EcoliCore>

⁴En <https://ecocyc.org/overviewsWeb/cel10v.shtml?orgid=ECOLI> se puede encontrar el mapa del genoma completo

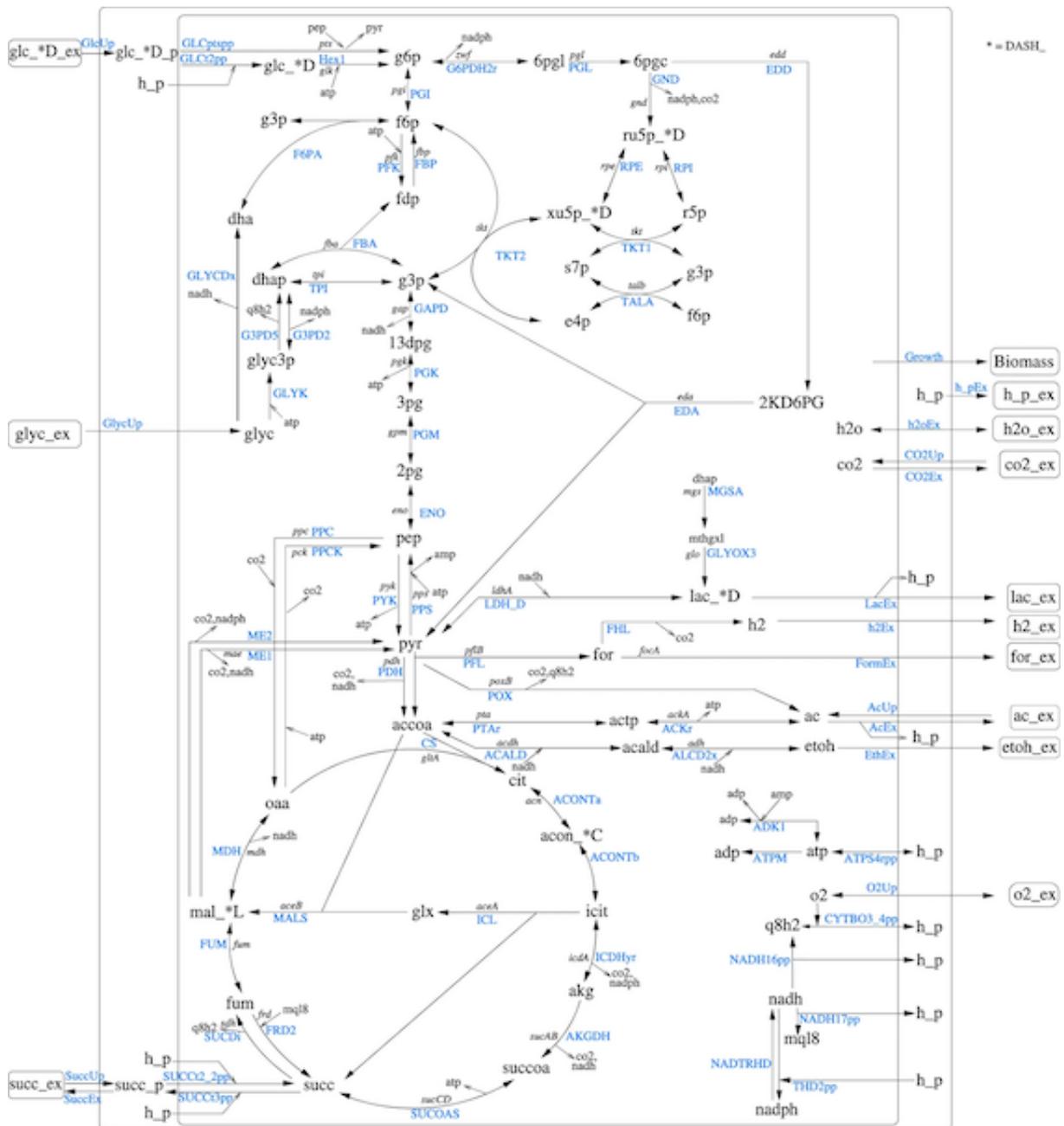


Figura 3.3: Imagen tomada de [18]. Modelo núcleo E. coli. La matriz estequiométrica del modelo y las restricciones de LB y UB se pueden descargar de la siguiente dirección: <http://systemsbiology.ucsd.edu/Downloads/EcoliCore>

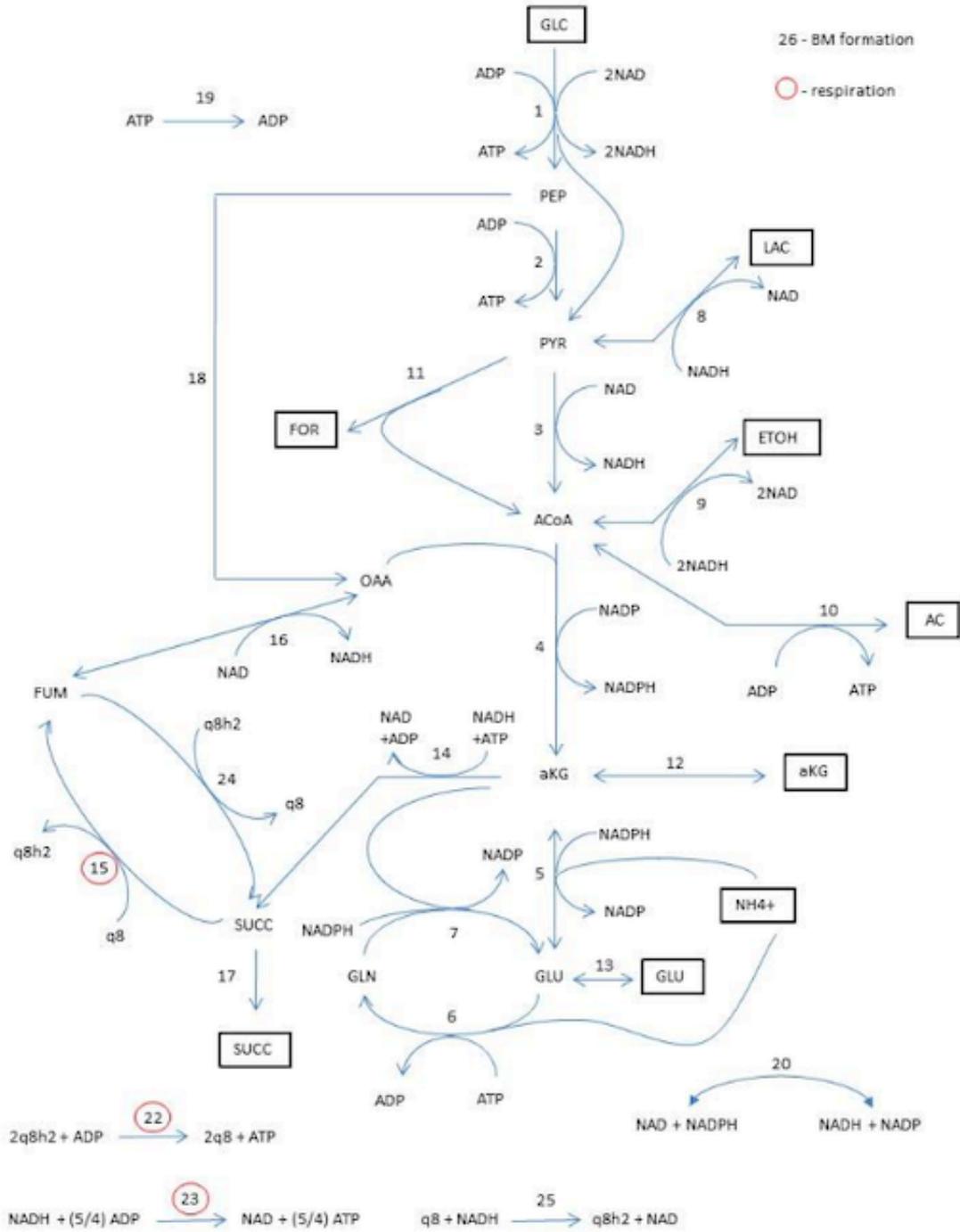


Figura 3.4: Modelo compacto *E. coli*. Tomado de [34]

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 | V22 | V23 | V24 | V25 | VBM |
|-------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|-----|-----|--------|
| ACoA | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | -1 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -3.748 |
| ADP | -1 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | -1.25 | 0 | 0 | 59.81 |
| aKG | 0 | 0 | 0 | 1 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1182 |
| ATP | 1 | 1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | 1.25 | 0 | 0 | -59.81 |
| FUM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 |
| GLN | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.256 |
| GLU | 0 | 0 | 0 | 0 | 1 | -1 | 2 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -4.941 |
| NAD | -2 | 0 | -1 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 0 | 0 | 0 | -1 | 0 | 1 | 0 | 1 | -3.547 |
| NADH | 2 | 0 | 1 | 0 | 0 | 0 | 0 | -1 | -2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | -1 | 3.547 |
| NADP | 0 | 0 | 0 | -1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 13.028 |
| NADPH | 0 | 0 | 0 | 1 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | -13.03 |
| OAA | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -1.787 |
| PEP | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -0.519 |
| PYR | 1 | 1 | -1 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -2.833 |
| q8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | -1 | 0 |
| q8h2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | -1 | 1 | 0 |
| SUCC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Tabla 3.3: Modelo compacto E. coli. Matriz estequiométrica

| | LB | UB |
|-----|-------|------|
| V1 | 0 | 10 |
| V2 | 0 | 1000 |
| V3 | 0 | 1000 |
| V4 | 0 | 1000 |
| V5 | -1000 | 1000 |
| V6 | 0 | 1000 |
| V7 | 0 | 1000 |
| V8 | 0 | 1000 |
| V9 | 0 | 1000 |
| V10 | 0 | 1000 |
| V11 | 0 | 1000 |
| V12 | 0 | 1000 |
| V13 | 0 | 1000 |
| V14 | 0 | 1000 |
| V15 | 0 | 1000 |
| V16 | -1000 | 1000 |
| V17 | 0 | 1000 |
| V18 | 0 | 1000 |
| V19 | 8.39 | 1000 |
| V20 | -1000 | 1000 |
| V22 | 0 | 1000 |
| V23 | 0 | 1000 |
| V24 | 0 | 1000 |
| V25 | 0 | 1000 |

Tabla 3.4: Modelo compacto E. coli. Restricciones

Capítulo 4

Algoritmos implementados

En esta sección se presenta una introducción de los algoritmos de los tres métodos de aproximación de volumen que fueron implementados en este trabajo.

En la Sección 1 se explica el método desarrollado en [34], que se basa en el concepto de Monte Carlo Multi Face. Se intenta explicar qué tiene este método en común con Monte Carlo directo, cómo se puede encontrar el espacio de referencia que se necesita y cómo a partir de este espacio y un conjunto de puntos sorteados dentro del mismo es que se llega a la aproximación del volumen de P .

En la Sección 2 se presenta el método propuesto por Fok y Crevier (FC) en [14].

La Sección 3, presenta los conceptos que permiten entender el estimador basado en MCMC presentado en [21]. Se explica cómo utilizando la técnica de integración termodinámica, que se utiliza en física y química, se puede llegar a la estimación del volumen de un politopo P . Para entender esta técnica también es necesario explicar el algoritmo Metropolis Hasting (MH) que permite sortear muestras a partir de una función de probabilidad desconocida. Y luego, se explica la técnica de parallel tempering (PT) que mejora el uso de MH en espacios de altas dimensiones.

Finalmente, como todos los algoritmos estudiados en este trabajo tienen en común que necesitan de un punto inicial perteneciente al politopo, en la Sección 4 se describe como encontrar dicho punto. También todos estos algoritmos funcionan con politopos de dimensión completa, por lo que, en la Sección 5, describimos un método que permite transformar politopos que no cumplen esta condición, a través de una transformación lineal en un politopo de dimensión completa, obteniendo una relación entre los volúmenes de ambos.

4.1. Calculo de Volumen sorteando muestras dentro del politopo

Para poder estimar el volumen de un politopo P aplicando Monte Carlo directo, primero es necesario relacionar el volumen de P , con el volumen de un espacio de referencia R . Dentro de R , se toman K muestras aleatorias y luego considerando la cantidad de muestras M que caen dentro de P , cuando K tiende a infinito, se obtiene un estimador del volumen:

$$vol(P) \approx vol(R) \frac{M}{K}$$

El espacio de referencia debe cumplir con dos características:

1. Contener completamente al politopo
2. Contar con una forma accesible de calcular su volumen.

Para obtener dicho espacio, en [34] se propone encontrar el hiper-rectángulo que más se ajusta al espacio de soluciones mediante la resolución de un conjunto de problemas de optimización. Con esto se busca hacer los muestreos de Monte Carlo lo más eficiente posible, reduciendo al máximo el espacio de referencia y al mismo tiempo haciendo que calcular su volumen sea muy simple. Como se trata de un hiper-rectángulo el volumen del espacio de referencia es el producto de los lados.

Primero es necesario encontrar una base ortonormal del espacio nulo ($Nul(A) = x \in R^n : Ax = 0$) de la

matriz estequiométrica, que se utiliza para proporcionar una base ortonormal del subespacio donde esta incluido el espacio de referencia buscado. Para encontrar una base del espacio nulo de una matriz A , el primer paso es obtener la descomposición QR de A , que consiste en la descomposición de una matriz en dos matrices: una matriz ortogonal Q y una matriz triangular superior R . A partir de las últimas columnas de la matriz Q , se obtiene una base ortonormal del espacio nulo de A . La cantidad de columnas de la matriz Q que se deben tomar corresponde a la dimensión del espacio nulo de la matriz. Las últimas columnas de la matriz Q , entonces, conforman la matriz W , cuyas columnas corresponden a la base ortonormal del espacio nulo de A . Multiplicando los vectores de esta base por escalares aleatorios, se obtienen puntos que pertenecen al espacio de soluciones, y puede posteriormente controlarse si estos puntos cumplen o no con las restricciones que lo definen.

Finalmente, para encontrar la región de referencia, es necesario resolver dos problemas de programación lineal por cada variable en W (uno de minimización y otro de maximización), en los que las restricciones implican que se cumplan las restricciones que definen a P .

Parámetros: $W \in R^{m \times n}, b \in R^n$
 Variables: $x_i \forall i = 1..n$
 Función objetivo: *Maximizar/Minimizar* x_i
 Restricciones: $\sum_{j=1}^n W_{ij}x_j \leq b_i \forall i = 1..n$

Una vez que se obtienen las cotas inferior y superior (x^{min}, x^{max} respectivamente) a partir de la resolución de estos problemas de programación lineal se puede construir la región de referencia como el conjunto de puntos que satisfacen: $I \leq x^{max}$ y $I \geq x^{min}$, en donde I es la matriz identidad y x^{max}, x^{min} son las cotas superior e inferior obtenidas.

El problema de aplicar Monte Carlo directo para aproximar el volumen de P , por más que se tenga este espacio de referencia más ajustado al politopo, es que el método es ineficiente para aproximar el volumen en espacios de altas dimensiones. A medida que aumenta la dimensión, también aumenta la diferencia de volumen entre el espacio de referencia y el espacio de soluciones, por consiguiente se hace difícil sortear muestras que caigan dentro de este espacio. Por eso es que en [34] se desarrolla una metodología para aproximar el volumen, que utiliza caminatas aleatorias para sortear muestras dentro del politopo. Este algoritmo calcula el volumen expresado como un producto de estimaciones de relaciones de volúmenes igual que en el enfoque Monte Carlo Multifase. Primero, se determina el espacio de referencia como se describió anteriormente. Una vez que se tiene el espacio de referencia, las restricciones que definen el politopo se van agregando una a una, y se van generando muestreos dentro de estas cajas en cada iteración. Teniendo esta serie de muestreos, y conociendo el volumen del espacio de referencia del que se parte se puede calcular el estimador del volumen del espacio de soluciones. Esto se hace contabilizando en cada iteración, qué fracción de los puntos muestreados cumple con la siguiente restricción, y al final multiplicar todas estas fracciones por el volumen de la caja de la que se parte.

En el anexo D se encuentra un ejemplo aplicando este algoritmo al modelo metabólico simple presentado en 3.3.1

4.2. Algoritmo Fok-Crevier

Como mencionamos anteriormente en [26], se hace un estudio de la aplicación del método desarrollado por Fok y Crevier al caso real del modelo Núcleo E. coli. A continuación daremos una descripción mas detallada de dicho método.

Fok y Crevier [14] propusieron un método directo de Monte Carlo para la estimación del volumen en espacios de alta dimensión, suponiendo que es posible encontrar para cada dirección e , el punto en el que una línea recta que comienza desde un punto interior del dominio con dirección e golpea el límite. Este método se puede aplicar al cálculo del volumen V para cualquier región R , pero en esta tesis nos limitamos a su uso en regiones convexas como los politopos.

El estimador FC utiliza la hiperesfera de radio r_s y centro C , como la región de referencia R_s que contiene a P . Lo que se logra utilizando esta región de referencia es que si $\lambda(e)$ es la distancia entre C y el borde P , entonces la probabilidad de que un punto aleatorio x dentro de R_s , en dirección e , se encuentre dentro de P es:

$$Prob[x \in P|e] = [\lambda(e)/r_s]^d \quad (4.1)$$

donde d es la dimensión de P .

Esto quiere decir que para todos los puntos en dirección e , la fracción esperada de puntos que pertenecen a P

es igual al resultado de esta ecuación. Al mismo tiempo tenemos que la proporción del volumen V de P sobre el volumen V_s de R_s , es la fracción esperada de puntos dentro de P cuando estos puntos están uniformemente distribuidos. Entonces, la probabilidad de que un punto x , pertenezca a P es igual a $P(x \in P|e)$ por la probabilidad de seleccionar la dirección e a partir de la distribución uniforme.

Por lo tanto se cumple que:

$$Prob[x \in P] = V/V_s = \int_{\forall e} [\lambda(e)/r_s]^d f_0(e) \quad (4.2)$$

Donde:

$f_0(e) = 1/V_0$ es la función de densidad de probabilidad de las direcciones e distribuidas uniformemente. V_0 es la superficie de la hiper-esfera d -dimensional de radio uno.

V_s y V_0 están dadas por:

$$V_s = Kr_s^d \text{ con } K = \begin{cases} \pi^m/m! & \text{cuando } d = 2m \\ (\pi^m)(m!)(2^d)/d! & \text{cuando } d = 2m + 1 \end{cases}$$

$$V_0 = dK$$

Sustituyendo V_s en 4.2, tenemos que:

$$V = K \int_{\forall e} [\lambda(e)]^d f_0(e) dz$$

$$\hat{V} = E[K\lambda(Z)^d]$$

$$\hat{V} = K \left[\frac{1}{k} \sum_{i=1}^k (\lambda^{(i)})^d \right]$$

donde Z es una variable aleatoria distribuida uniformemente como $f_0(z)$, $\lambda^{(i)} = \lambda(z^{(i)})$, $z^{(i)}$ es la i -ésima dirección muestreada desde $f_0(z)$

El punto inicial x_0 puede ser cualquier punto que pertenezca a P , pero dado que la elección de este punto afecta la función de distancia $\lambda(x)$, dicha elección afecta también la varianza del estimador. Una buena elección de x_0 puede hacer que los valores de $\lambda(x)$ sean lo más similares posibles, con lo cual se obtendría un varianza pequeña.

Se puede implementar otra variante de este algoritmo, si en lugar de utilizar siempre el mismo punto inicial para calcular las distancias, en cada iteración sorteamos una dirección, calculamos las distancias y calculamos un nuevo punto perteneciente a P para utilizar como punto inicial en la siguiente iteración.

Como x_0 puede ser cualquier punto dentro de P , su elección puede seguir cualquier distribución, si se utiliza hit-and-run para hacer esta selección se obtiene una distribución asintóticamente uniformemente distribuida.

En el anexo G.2 se encuentra el pseudocódigo del algoritmo para sus dos variantes.

Como ya se dijo en secciones anteriores, el problema que tiene aplicar Monte Carlo directo al cálculo del volumen en espacios de altas dimensiones es que cuando aumenta la dimensionalidad, el volumen del espacio aumenta exponencialmente haciendo que sea muy difícil sortear un punto, dentro del espacio de referencia que caiga dentro de P . Este método no tiene ese problema, porque siempre se están sorteando puntos dentro de P , pero con el aumento de dimensión, igual que en MC, se pierde calidad del estimador debido a la variación en la distribución de las distancias entre un punto interior y el borde del politopo en una dirección e .

4.3. Metodo basado en MCMC

Como mencionamos anteriormente en [26], se hace un estudio de la aplicación de un método basado en Markov Chain Monte Carlo (MCMC) al caso real del modelo Núcleo E. coli. Los métodos de MCMC construyen una cadena Markov con un muestreo de variables dependientes que tienen una distribución estacionaria $p(x)$, lo que hace que podamos extraer muestras de una distribución, incluso si no podemos calcularla. Primero damos una breve descripción del método Metropolis-Hastings, que es un algoritmo que permite simular una cadena de Markov con distribución estacionaria $p(x)$. A continuación se explica el método desarrollado en [26].

4.3.1. Metropolis-Hastings

Los siguientes conceptos fueron estudiados en [28]. Esta técnica requiere una distribución simple llamada distribución de proporción o modelo de transición, $q(x' \leftarrow x)$ para ayudar a extraer muestras de una distribución

posterior que no conocemos $p(x)$. El algoritmo usa q para caminar aleatoriamente en el espacio de distribución, aceptando o rechazando saltos a nuevas posiciones según la probabilidad de la muestra.

Algorithm 1: Metropolis-Hastings

Entradas: estado inicial x , número de iteraciones S ;

for $s = 1 \dots S$ **do**

1. Propuesta $x' \sim q(x' \leftarrow x)$;
2. Calcular $a = \frac{p(x')q(x \leftarrow x')}{p(x)q(x' \leftarrow x)}$;
3. Setear $x = x'$ con probabilidad $\min(1, a)$.
 - a. Sortear $r \sim U[0, 1]$;
 - b. **if** $r < a$ **then** $x \leftarrow x'$;

end

La configuración de x al final de cada iteración se considera como una muestra de la distribución de probabilidad objetivo $p(x)$. Las muestras adyacentes son idénticas cuando el estado no se actualiza en el paso 3b), pero cada iteración debe registrarse como parte de la cadena de Markov.

El algoritmo es válido para cualquier distribución de propuesta q . Muchas veces, esta es una distribución simétrica (por ejemplo una distribución normal), es decir, $q(x \leftarrow x') = q(x' \leftarrow x)$, que hace que la relación de aceptación en el paso 2. se simplifique a una relación bajo la distribución de interés, $a = p(x')/p(x)$.

El paso 3 de aceptar/rechazar en el algoritmo corrige la falta de coincidencia entre las distribuciones propuestas y de destino. Esto significa que si un x es más probable que el actual x , siempre aceptamos x . Si es menos probable que el x actual, entonces podríamos aceptarlo o rechazarlo al azar con una probabilidad decreciente, cuanto menos probable sea.

En el anexo F se puede encontrar un ejemplo en una dimensión para ilustrar MH (tomado de [10]).

Gracias a los métodos de MCMC, se puede simular a partir de un modelo de probabilidad complejo $p(x)$ sin conocer su constante de normalización. Es decir, uno puede evaluar $f(x)$, una función de densidad no normalizada, pero no puede calcular directamente $z = \int f(x)p(x)dx$, la constante de normalización.

Las distribuciones para las cuales $f(x)$ puede calcularse fácilmente pero z es intratable surgen en muchos modelos estadísticos, como modelos espaciales o modelos jerárquicos bayesianos. También en muchos casos, se expresa una cantidad de interés como una constante normalizadora de una densidad a partir de la cual se pueden realizar sorteos con el fin de tener un problema de este estilo y mediante su resolución poder obtener el valor de esta cantidad. Más adelante veremos que este es el enfoque que tiene el método de cálculo de volumen que vamos a describir a continuación.

4.3.2. Estimación del volumen utilizando integración termodinámica

La técnica de integración termodinámica (también conocida como Path Sampling) [25] permite realizar una evaluación numérica de la relación logarítmica entre dos constantes de normalización Z_0, Z_1 , mediante la siguiente ecuación:

$$\ln(Z_1/Z_0) = \ln(Z_1) - \ln(Z_0) = \int_0^1 \frac{\partial \ln(Z_\beta)}{\partial \beta} d\beta \quad (4.3)$$

Para esto se consideran $q_i(x)$ con $i = 0, 1$ como dos funciones de densidades no normalizadas definidas en el mismo espacio de parámetros .

Las densidades de probabilidad verdaderas correspondientes se describen por

$$p_i(x) = q_i(x)/Z_i \quad i = 0, 1 \quad (4.4)$$

en donde los Z_i son las constantes de normalización.

$$Z_i = \int_{\Theta} q_i(x) dx \quad i = 0, 1 \quad (4.5)$$

Para hacer esto, se define una ruta continua y diferenciable $(q_\beta)_{0 \leq \beta \leq 1}$ en el espacio de densidades no normalizadas, uniendo q_0 y q_1 .

Por extensión, para cualquier β , $0 \leq \beta \leq 1$, p_β y Z_β se definen como:

$$p_\beta(x) = q_\beta(x)/Z_\beta \quad (4.6)$$

$$Z_\beta = \int_{\Theta} q_\beta(x) dx \quad (4.7)$$

La idea clave de la integración termodinámica es que, para cualquier valor de β entre 0 y 1, se puede ejecutar una cadena de Markov en la que q_β se utiliza como la densidad no normalizada en la relación Metropolis-Hastings. Por definición, esto produce una muestra de valores de parámetros extraídos de la distribución de probabilidad p_β . Finalmente, este cálculo se puede realizar para una serie de valores de β regularmente espaciados entre 0 y 1, lo que implica ejecutar una cadena de Markov para cada valor de β . Luego se utilizan los valores estimados para aproximar la integral sobre $[0,1]$ en (4.3) utilizando algún método de aproximación numérico.

En el método basado en MCMC de [26] se expresa el volumen de un politopo como una constante normalizadora de una densidad. Como la función de distancia $\lambda(x)$ introducida en el método de FC es positiva por construcción, se propone interpretar al integrando en (4.3) como una densidad de probabilidad normalizada por el volumen V .

$$f(x) = \frac{1}{V} \frac{\lambda^d(x)}{d} \quad (4.8)$$

En donde $\lambda(x)$ es la distancia entre un punto interior a P y el borde del mismo en la dirección x y d es la dimensión de P .

Luego se utiliza la técnica de integración termodinámica (path sampling) para estimar el ratio entre dos constantes de normalización.

Se define:

$$f(x) = \frac{\lambda^d(x)}{d} \quad (4.9)$$

$$q_i(x) = f^\beta(x) \quad (4.10)$$

Entonces aplicando el método de integración termodinámica tenemos que:

$$\begin{aligned} p_\beta &= \frac{1}{Z_\beta} f^\beta(x) = \frac{1}{Z_\beta} \exp(\beta \ln(f(x))) \\ Z_\beta &= \int_{\partial B_d} \exp(\beta \ln(f(x(\omega)))) d\omega \\ Z_0 &= A_{B_d} = \frac{2\pi^{d/2}}{\Gamma(d/2)} \\ Z_1 &= V \\ \ln\left(\frac{Z_1}{Z_0}\right) &= \int_0^1 \frac{\partial \ln(Z_\beta) \partial \beta}{\partial \beta} \\ &= \int_0^1 \int_{\partial B_d} \ln(f(x(\omega))) p_\beta(x(\omega)) \partial \omega \partial \beta \\ &= \int_0^1 E_{p_\beta}[\ln(f)] \partial \beta \end{aligned} \quad (4.11)$$

Para $\beta = 0$, las direcciones resultantes se distribuyen uniformemente en la esfera unitaria B_d , lo que significa que la probabilidad para cada dirección es la misma.

En el caso de $\beta = 1$, las muestras se ubican preferiblemente cerca de los vértices del politopo P , dando a la dirección e una mayor probabilidad de estar en la dirección de un vértice.

Z_0 es la superficie de la unidad N -dimensional de la hiperesfera.

La idea del método para estimar el volumen es, aplicar MH para obtener muestras x_k de las densidades p_β y así obtener un estimador MCMC de E_{p_β} que depende de beta y el número de muestras K .

$$\lim_{K \rightarrow +\infty} \hat{E}_K(\beta) = \lim_{K \rightarrow +\infty} \frac{1}{K} \sum_k \beta \ln(f(x_k)) = E_{p_\beta}[\ln(f)] \quad (4.12)$$

Aplicando el método del trapecio¹, obtenemos el estimador correspondiente para el volumen del politopo P .

$$\hat{V} = Z_0 \exp\left(\Delta\beta \left(\frac{\hat{E}_0 + \hat{E}_1}{2} + \sum_{j=1}^{n-1} \hat{E}_{j\Delta\beta}\right)\right)$$

¹https://es.wikipedia.org/wiki/Regla_del_trapecio

con subintervalos de igual tamaño $\Delta\beta = 1/b$ para un determinado número b .

Luego se mejora la precisión del método aplicando la técnica de parallel tempering (PT) [30], aprovechando que esta técnica se basa en simular M réplicas del sistema de interés original.

Para funciones de densidad de probabilidad con picos bruscos p_β , las cadenas de Markov tienden a quedarse trancadas en los estados con probabilidades más altas. Para evitarlo el algoritmo de cálculo de volumen incluye la implementación de la técnica parallel tempering, intercambiando estados entre las cadenas que se utilizan para estimar cada p_β .

La idea básica de PT es ejecutar más de una cadena a diferentes “temperaturas inversas” $0 = \beta_1 < \beta_2 < \dots < \beta_n = 1$. Esta técnica no está necesariamente relacionada con la integración termodinámica, pero en este caso las “temperaturas inversas” β son las mismas que las definidas para este método anteriormente.

Una secuencia con un valor bajo de β explora todo el espacio de parámetros, mientras que una cadena que corre con un valor alto de β tiende a ubicarse cerca de los picos. Entonces en cada paso que se genere un nuevo estado

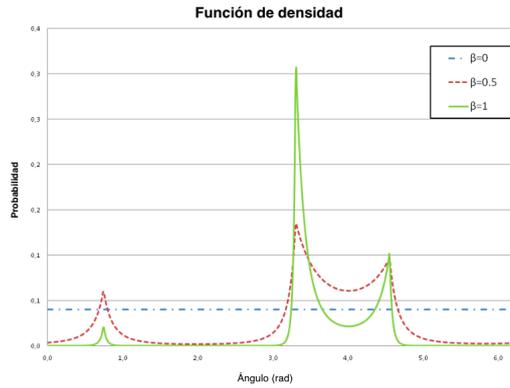


Figura 4.1: Imagen tomada de [26]. Densidades de probabilidad p_β para diferentes temperaturas inversas β mostradas en función del ángulo de dirección de búsqueda. A valores bajos de β , las muestras se distribuyen uniformemente en la esfera de la unidad, mientras que a altas temperaturas las distribuciones β alcanzan un pico máximo. Esto muestra que es poco probable que se extraigan muestras en todas las regiones con picos sin aplicar PT.

en la cadena de Markov, se introduce una regla de intercambio que permite un intercambio de los estados $x_k^{(i)}$ entre diferentes temperaturas β_k . Los dos estados $x_k^{(i)}$ y $x_k^{(j)}$ se intercambiarán con la probabilidad:

$$\min\{1, \exp((\beta_i - \beta_j)(g(x_k^{(i)}) - g(x_k^{(j)})))\}$$

Esta regla permite una exploración más rápida de todos los picos a valores β altos, haciendo que la Cadena de Markov salte de pico a otro .

Por lo general cuando se utiliza PT, la función de densidad de interés es cuando $\beta = 1$, en este caso se utilizan todas las temperaturas para la integración numérica del estimador de volumen, por lo que, se aprovecha el conjunto completo de muestras calculadas.

4.4. Punto inicial

Los tres métodos que se van a implementar necesitan como punto inicial un punto interior del politopo. Este punto se puede obtener fácilmente, en espacios de bajas dimensiones sorteando puntos aleatorios y verificando si el punto cumple con todas las restricciones que definen al politopo. Pero a medida que la dimensión crece, la probabilidad de que un punto sorteado aleatoriamente caiga dentro del politopo se hace cada vez más pequeña. Por este motivo, para encontrar el punto interior planteamos el siguiente problema de optimización:

Parámetros: $A \in R^{m \times n}$, $b \in R^n$

Variables:

- $\epsilon_i \forall i = 1..m$
- $x_i \forall i = 1..n$
- ξ

Función objetivo: Maximizar ξ

Restricciones:

- $\sum_{j=1}^{n-1} A_{ij}x_j + \epsilon_i = b_i \forall i = 1..n$
- $\xi \leq \epsilon_i \forall i = 1..n$
- $\epsilon_i \geq 0 \forall i = 1..n$

De esta manera lo que se busca es encontrar un punto que cumpla con todas las restricciones, por lo que va a ser un punto interior, y que la distancia de ese punto a cada hyper plano que define al politopo sea la máxima posible. Con esto se busca que el punto seleccionado esté lo más lejos posible de los bordes de P .

Otra opción para obtener un punto interior dentro de P es utilizar el centro de la bola de Chebychev ², es decir, la bola inscrita más grande en P . Para esto, se resuelve el siguiente problema de programación lineal:

$$\begin{cases} \text{maximizar } R \\ \text{s.t.} \\ A_i x + R \|A_i\|_2 \leq b_i, i = 1, \dots, m \\ R \geq 0 \end{cases}$$

donde S_i es la i -ésima fila de A , y los valores óptimos de R y $x \in R^d$ producen, respectivamente, el radio r y el centro c de la bola de Chebychev.

4.5. Restricciones con igualdades

Cuando en la definición de P existe alguna restricción de igualdad, tenemos que la dimensión es menor que la cantidad de variables que componen el conjunto de restricciones. Como los métodos aleatorios se basan en el muestreo en un politopo de dimensión completa (porque la probabilidad de muestreo dentro de espacios de solución degenerados es 0), en estos casos es necesario hacer un pre-procesamiento de P , antes de poder realizar la estimación del volumen.

Para esto, seguimos la metodología presentada en [35]. En este trabajo se define la forma compacta de un conjunto de restricciones de la forma $Ax \leq b$ y luego se plantea y demuestra un teorema que permite obtener el volumen del espacio de soluciones a partir de esta.

Definición 4.5.1. Forma compacta:

$$\begin{cases} x_p = A_p x_s + b_p & \text{proyeccion} \\ A_s x_s \leq b_s & \text{soporte} \end{cases}$$

donde $x = x_p \cap x_s$ y la parte de soporte $A_s x_s \leq b_s$ es de dimensión completa.

Dado $Ax \leq b$, se puede encontrar su forma compacta, mediante los siguientes pasos:

Se define una ecuación $a_i^T x \leq b_i$ como implícita con respecto a $Ax \leq b$ si y sólo si considerando todas las ecuaciones pertenecientes al sistema implica que $a_i^T x = b_i$. Por ejemplo si $Ax \leq b$ contuviera las ecuaciones $x \leq y$ y $y \leq x$, ambas serían ecuaciones implícitas ya que esto implica que $x = y$. Para identificar este tipo de ecuaciones se puede introducir una variable de holgura por cada ecuación que conforma el sistema y resolver un problema de optimización por cada una de ellas que las maximice. En un problema de optimización, una variable de holgura es una variable que se suma a una restricción de desigualdad para transformarla en una igualdad. La introducción de una variable de holgura reemplaza una restricción de desigualdad con una restricción de igualdad y una restricción de no negatividad en la variable de holgura.

1. Agregar variables de holgura a todas las ecuaciones. Resolver los problemas de optimización para maximizarlas. Si hay un variable de holgura, cuyo máximo es igual a cero, quiere decir que la restricción que la involucra es una ecuación implícita.
2. Reescribir todas las ecuaciones implícitas como ecuación.
3. Como $\dim(x_p) \leq \dim(x_s)$ se puede aplicar la eliminación Gaussiana en el conjunto de ecuaciones y remover ecuaciones triviales, para luego tener el conjunto de ecuaciones en forma escalonada reducida.
4. Determinar las variables libres del sistema de ecuaciones y escribir todas las demás en función de estas. Las ecuaciones quedan con la forma $x_p = A_p x_s + b_p$. Las variables libres son las componentes de x_s , y las demás componen x_p .

²https://en.wikipedia.org/wiki/Chebyshev_center

Si $Ax \leq b$ ya es full-dimensional, la forma compacta es igual y la parte de proyección es vacía. El volumen de la parte de soporte, puede ser aproximado aplicando los métodos aleatorios. La relación entre el volumen de $Ax \leq b$ y la parte de soporte está dada por la siguiente ecuación:

Teorema 4.5.1 (Volumen). Sea $Ax \leq b$ un conjunto de ecuaciones lineales y $(x_p = A_p x_s + b_p) \wedge (A_s x_s \leq b_s)$ su forma compacta, entonces:

$$\text{Vol}(Ax \leq b) = \sqrt{\det(I + A_p^T A_p)} \text{Vol}(A_s x_s \leq b_s)$$

La demostración de este teorema se encuentra en [35]. En el anexo E se encuentra un ejemplo aplicando esta metodología al modelo simple.

4.6. Detalles de la implementación

El desarrollo y las pruebas en este trabajo se realizaron en un computador personal con las siguientes características:

- System Version: macOS Catalina version 10.15.6
- MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)
- Processor: 2.5 GHz Dual-Core Intel Core i7
- Memory: 16 GB 2133 MHz LPDDR3
- Graphics: Intel Iris Plus Graphics 640 1536 MB

Para la implementación se utilizó el lenguaje de programación **Python** version 3.7.3

Para la resolución de problemas de programación lineal se utilizó la librería **PuLP**³, versión 2.2. PuLP es un modelador para problemas de programación lineal escrito en Python que puede generar archivos LP y llamar a diferentes paquetes de resolución de problemas lineales, entre ellos **GLPK**⁴, que es el que se utiliza en esta implementación.

Para la simplificación de sistemas de ecuaciones lineales se utilizó la librería **Sympy**⁵ versión 1.5.1. Sympy permite hacer operaciones analíticas o con símbolos en lugar de con valores numéricos.

En el caso de la implementación del algoritmo para aproximar el volumen, utilizando puntos sorteados dentro del politopo, sección 4.1, se utilizaron las caminatas aleatorias RDHR, CDHR, BaW y BiW proporcionadas por la librería **VolEsti**. Para la integración con estos metodos, se utilizó la interfaz que viene incluida en dicha librería. Esta interfaz esta implementada con Cython⁶, que es un compilador estático de optimización para el lenguaje de programación Python. Utilizando esta herramienta se puede escribir extensiones C para Python.

En la implementación del algoritmo basado en MCMC, sección 4.3 se utilizaron las librerías de Python, **Theano** versión 1.0.4 y **PyMC3** version 3.9.2.

PyMC3 es un framework open source de programación probabilística escrito en Python que ofrece una implementación flexible y extensible que se utilizó para facilitar la implementación de MCMC. Este framework usa Theano⁷ para compilar programas probabilísticos sobre la marcha en C para aumentar la velocidad. Theano es una biblioteca de Python que le permite definir, optimizar y evaluar expresiones matemáticas que involucran matrices multidimensionales de manera eficiente.

El código fuente de PyMC3 está alojado en GitHub en <https://github.com/pymc-devs/pymc3> y se distribuye bajo la liberal ApacheLicense2.0. La documentación completa está disponible en <http://pymc-devs.github.io/pymc3/>.

Debido a que PyMC permite la creación de distribuciones de probabilidad definidas por el usuario, se definió un modelo, en el cual las variables aleatorias siguen las distribuciones p_β . En este modelo, definimos una variable aleatoria por cada valor de β e implementamos la función de distribución que depende de la distancia del punto inicial a los bordes del politopo, en Theano. Cabe destacar que por un tema de estabilidad numérica, PyMC

³<https://github.com/coin-or/pulp>

⁴<https://www.gnu.org/software/glpk/>

⁵<https://www.sympy.org/en/index.html>

⁶<https://cython.org>

⁷<http://deeplearning.net/software/theano/>

espera que lo que devuelva la implementación de la función de probabilidad sea el logaritmo de probabilidad $\ln(p_\beta)$. Esto con el objetivo de evitar las divisiones en el paso de Metropolis Hastings.

Algorithm 2: Modelo PyMC3

```

with pm.Model() as model:
for i in range(b + 1): do
  |  $x_i = \text{CustomDensityDist}(\text{polytope}=P, \text{center}=x_0, \text{beta}=i / b, \text{name}='x_i')$ 
end
trace = pm.sample(draws=n, init= $x_0$ , step=CustomMetropolis(proposal=CustomProposal),
  callback=swapping(model))

```

Una vez que se tiene el modelo definido, el siguiente paso es extraer muestras de las variables aleatorias a partir de la distribución posterior utilizando métodos de muestreo MCMC. Esto se hace mediante el método `sample` proporcionado por PyMC3. Para realizar este muestreo, se especifica un método implementado en Theano que corresponde a una única iteración de un algoritmo de MCMC particular, como Metrópolis. En este caso, implementamos el método `CustomMetropolis`, que extiende al método `Metropolis` proporcionado por PyMC3, y que incorpora la función de proporción definida para este algoritmo, `CustomProposal`. Dicha función sortea de manera uniforme un vector unitario en la esfera de radio uno. La función `sample` ejecuta el método que se le pasa en el parámetro `step` para el número de iteraciones dado y devuelve un objeto `trace` que contiene las muestras recolectadas, en el orden en que fueron recolectadas. Mediante el parámetro `init` se le pasa el punto inicial $x_0 \in P$ que se utilizan como punto de partida para el muestreo y la optimización de forma predeterminada. Finalmente en el parámetro `callback`, pasamos una función implementada en Theano que es ejecutada al finalizar cada paso de Metropolis que es ejecutado. Dicha función se encarga de realizar el intercambio entre las variables aleatorias definidas, que representa el intercambio entre cadenas realizado en el método de Parallel tempering.

Por otra parte, cabe destacar que uno de los objetivos del diseño de Theano es especificar cálculos a un nivel abstracto, de modo que el compilador de funciones internas tenga mucha flexibilidad. Una de las formas en que se puede aprovechar la flexibilidad es realizando cálculos en una tarjeta gráfica. Por lo que Theano permite mediante una simple configuración, el uso de la GPU (Unidad de Procesamiento de Gráficos) para realizar los cálculos. Una GPU es un procesador que se encarga de procesar todos los gráficos que utiliza el sistema de computo. Debido a que estos procesadores tienen bajo precio en relación a su potencia de cálculo, y una gran capacidad de paralelismo, se consideran atractivas para su uso en aplicaciones fuera de los gráficos por computadora, especialmente en el ámbito científico y de simulación.

Existen herramientas para facilitar el desarrollo de aplicaciones GPGPU⁸ (del inglés general-purpose computing on graphics processing units), al abstraer muchos de los detalles relacionados con los gráficos, y presentar una interfaz de más alto nivel. La opción más extendida en la actualidad es CUDA, de NVidia, una extensión de C que permite la codificación de algoritmos en GPU de NVidia. Por otra parte, está OpenCL, una combinación de interfaz y lenguaje de programación para el desarrollo de aplicaciones paralelas que puedan ser ejecutadas, de forma transparente, en diversas unidades de procesamiento.

Theano está diseñado para trabajar con CUDA y se supone que también lo está para trabajar con OpenCL, sin embargo, el soporte actual está incompleto.

En este trabajo logramos configurar Theano para que haga uso de la GPU, utilizando OpenCL, por no contar con una GPU NVidia que nos permite configurarlo para utilizarlo con CUDA. Sin embargo no fue posible hacer que la implementación de nuestro modelo funcione utilizando esta configuración, por lo que no pudimos comprobar si el uso de GPU mejora el rendimiento del algoritmo.

⁸https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units

Capítulo 5

Experimentación numérica

En este capítulo, la idea principal es evaluar el comportamiento de los diferentes métodos estudiados (tanto los que implementamos en este trabajo como los que están implementados en la librería de código abierto VolEsti) y determinar si es posible utilizarlos para estimar el volumen de los modelos metabólicos presentados en la sección 3.3. Para esto, en una primera etapa se realizaron pruebas aplicando los algoritmos a un conjunto de politopos de los cuales se conoce su volumen exacto (en el anexo C se puede consultarse una breve descripción de dichos modelos). Luego, a partir de los resultados obtenidos con estos modelos y teniendo en cuenta las características especiales de los modelos metabólicos estudiados, seleccionamos los algoritmos que vamos a aplicar en dichos modelos.

Para evaluar los algoritmos, cada experimento lo repetimos 20 veces. El tiempo informado para cada experimento es la media de los 20 tiempos obtenidos en cada repetición. Los valores informados en la columna Volumen corresponden con el volumen exacto del politopo (calculado con VolEsti en todos los casos excepto en los politopos birkhoff cuyo volumen exacto fue tomado de [1]) Para el volumen estimado, se presentan los valores mínimos y máximos calculados a partir de las 20 repeticiones, la media μ (estimación del volumen) y la desviación estándar σ . En la columna $min \leq \mu \leq max$ se indica si el valor de la media μ se encuentra comprendido dentro de los valores mínimos y máximos. También incluimos en los resultados el porcentaje de error de aproximación medio $(vol(P) - \mu)/vol(P) \times 100$. Por otra parte, como la media es la estimación del volumen, y este es un valor positivo, se puede usar el coeficiente de variación para evaluar los diferentes resultados obtenidos dentro de las 20 repeticiones de cada experimento. El coeficiente de variación se calcula como la desviación estándar sobre la media $(\frac{\sigma}{\mu})$ y es una medida estadística que brinda información acerca de la dispersión relativa de un conjunto de datos. Esta medida toma valores entre 0 y 1. Si el coeficiente es próximo al 0, significa que existe poca variabilidad en los datos de la muestra. En cambio, si tienden a 1 es una muestra muy dispersa, es decir que los resultados de cada uno de los 20 experimentos fueron considerablemente diferentes entre sí, y la media pierde confiabilidad. Para facilitar el análisis se puede expresar el coeficiente como un porcentaje, multiplicando por cien. Cuando el coeficiente de variación supera el 25% concluimos que la media es poco representativa, por lo que, en estos casos, la estimación no es buena.

Finalmente, para evaluar las estimaciones obtenidas aplicando los diferentes algoritmos, definimos que una aproximación es aceptable si los resultados presentan un error relativo porcentual menor al 10% y un coeficiente de variación menor a 25%.

5.1. Resultados VolEsti

A continuación, en las tablas 5.1, 5.2 y 5.3, se muestran los resultados obtenidos al aplicar los algoritmos SoB, CG y CB implementados en la librería VolEsti. Los valores de N y ϵ utilizados, son los que fueron determinados experimentalmente en [11]. Las filas que se presentan en color verde son las filas de las estimaciones que cumplieron el criterio de aceptación que definimos (Error relativo porcentual menor al 10% y un coeficiente de variación menor a 25%).

| P | m | d | W | ϵ | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $\min \leq \mu \leq \max$ | % Error |
|----------|-----|-----|-------|------------|------------|----------|----------|----------|----------------|----------|----------|---------------------------|---------|
| birkhoff | 9 | 4 | 4 | 0.05 | 2.14e+00 | 1.12e+00 | 1.13e+00 | 3.21e-03 | 0.28 | 1.12e+00 | 1.13e+00 | SI | 0.16 |
| birkhoff | 16 | 9 | 10 | 0.05 | 4.03e+01 | 6.21e-02 | 6.21e-02 | 2.21e-04 | 0.36 | 6.17e-02 | 6.25e-02 | SI | 0.03 |
| birkhoff | 25 | 16 | 18 | 0.05 | 3.12e+02 | 1.41e-04 | 1.42e-04 | 1.90e-06 | 1.34 | 1.40e-04 | 1.49e-04 | SI | 0.54 |
| birkhoff | 36 | 25 | 28 | 0.05 | 1.76e+03 | 7.35e-09 | 7.54e-09 | 1.05e-10 | 1.39 | 7.39e-09 | 7.63e-09 | SI | 2.50 |
| birkhoff | 49 | 36 | 40 | 0.05 | 7.11e+03 | 5.64e-15 | 6.25e-15 | 6.15e-16 | 9.85 | 5.63e-15 | 6.86e-15 | SI | 10.73 |
| cc_8_10 | 70 | 8 | 1000 | 0.70 | 2.25e+01 | 1.57e+05 | 2.44e+06 | 1.10e+05 | 4.50 | 2.29e+06 | 2.67e+06 | SI | 1456.92 |
| cc_8_10 | 70 | 8 | 10000 | 0.70 | 6.13e+02 | 1.57e+05 | 2.44e+06 | 8.86e+04 | 3.64 | 2.28e+06 | 2.69e+06 | SI | 1454.20 |
| cc_8_10 | 70 | 8 | 9 | 0.05 | 7.96e+01 | 1.57e+05 | 2.42e+06 | 3.09e+04 | 1.27 | 2.37e+06 | 2.46e+06 | SI | 1444.56 |
| cc_8_11 | 88 | 8 | 1000 | 0.70 | 3.93e+01 | 1.39e+06 | 1.25e+08 | 1.06e+07 | 8.46 | 1.08e+08 | 1.43e+08 | SI | 8866.74 |
| cc_8_11 | 88 | 8 | 10000 | 0.70 | 4.71e+02 | 1.39e+06 | 1.26e+08 | 5.84e+06 | 4.64 | 1.15e+08 | 1.38e+08 | SI | 8935.00 |
| cc_8_11 | 88 | 8 | 9 | 0.05 | 1.73e+02 | 1.39e+06 | 1.23e+08 | 7.63e+06 | 6.20 | 1.04e+08 | 1.39e+08 | SI | 8743.69 |
| cc_8_5 | 10 | 8 | 1000 | 0.70 | 3.85e+01 | 1.74e-03 | 1.75e-03 | 7.91e-05 | 4.53 | 1.59e-03 | 1.92e-03 | SI | 0.60 |
| cc_8_5 | 10 | 8 | 10000 | 0.70 | 1.69e+02 | 1.74e-03 | 1.72e-03 | 6.91e-05 | 4.02 | 1.56e-03 | 1.81e-03 | SI | 1.10 |
| cc_8_5 | 10 | 8 | 9 | 0.05 | 4.93e+01 | 1.74e-03 | 1.74e-03 | 1.75e-05 | 1.01 | 1.70e-03 | 1.76e-03 | SI | 0.01 |
| cc_8_6 | 18 | 8 | 1000 | 0.70 | 1.14e+01 | 4.44e-01 | 4.36e-01 | 2.35e-02 | 5.39 | 3.89e-01 | 4.96e-01 | SI | 1.85 |
| cc_8_6 | 18 | 8 | 10000 | 0.70 | 3.05e+02 | 4.44e-01 | 4.43e-01 | 1.16e-02 | 2.62 | 4.14e-01 | 4.60e-01 | SI | 0.29 |
| cc_8_6 | 18 | 8 | 9 | 0.05 | 6.25e+01 | 4.44e-01 | 4.46e-01 | 7.31e-03 | 1.64 | 4.31e-01 | 4.61e-01 | SI | 0.33 |
| cc_8_8 | 40 | 8 | 1000 | 0.70 | 1.46e+01 | 7.84e+02 | 3.10e+03 | 7.89e+01 | 2.54 | 2.88e+03 | 3.24e+03 | SI | 295.92 |
| cc_8_8 | 40 | 8 | 10000 | 0.70 | 1.71e+02 | 7.84e+02 | 3.09e+03 | 9.40e+01 | 3.04 | 2.92e+03 | 3.27e+03 | SI | 293.94 |
| cc_8_8 | 40 | 8 | 9 | 0.05 | 7.70e+01 | 7.84e+02 | 3.07e+03 | 3.27e+01 | 1.07 | 3.04e+03 | 3.14e+03 | SI | 291.48 |
| cc_8_9 | 54 | 8 | 1000 | 0.70 | 1.61e+01 | 1.33e+04 | 1.96e+05 | 5.16e+03 | 2.63 | 1.89e+05 | 2.10e+05 | SI | 1368.86 |
| cc_8_9 | 54 | 8 | 10000 | 0.70 | 1.57e+02 | 1.33e+04 | 1.97e+05 | 7.68e+03 | 3.90 | 1.86e+05 | 2.20e+05 | SI | 1376.73 |
| cc_8_9 | 54 | 8 | 9 | 0.05 | 9.90e+01 | 1.33e+04 | 1.98e+05 | 1.38e+03 | 0.70 | 1.96e+05 | 2.00e+05 | SI | 1380.60 |
| ccp | 56 | 10 | 11 | 0.05 | 3.78e+01 | 2.31e+00 | 2.31e+00 | 3.00e-03 | 0.13 | 2.31e+00 | 2.32e+00 | SI | 0.03 |
| cross | 4 | 2 | 2 | 0.05 | 1.18e-01 | 2.00e+00 | 2.00e+00 | 3.46e-03 | 0.17 | 1.99e+00 | 2.00e+00 | SI | 0.00 |

| P | m | d | W | ϵ | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $\min \leq \mu \leq \max$ | % Error |
|----------|-------|-----|-----|------------|------------|----------|----------|----------|----------------|----------|----------|---------------------------|---------|
| cross | 16 | 4 | 4 | 0.05 | 9.85e-01 | 6.67e-01 | 6.67e-01 | 8.51e-04 | 0.13 | 6.65e-01 | 6.69e-01 | SI | 0.02 |
| cross | 32 | 5 | 6 | 0.05 | 2.10e+00 | 2.67e-01 | 2.67e-01 | 3.26e-04 | 0.12 | 2.66e-01 | 2.67e-01 | SI | 0.01 |
| cross | 64 | 6 | 7 | 0.05 | 5.10e+00 | 8.89e-02 | 8.89e-02 | 9.57e-05 | 0.11 | 8.88e-02 | 8.91e-02 | SI | 0.04 |
| cross | 256 | 8 | 9 | 0.05 | 4.04e+01 | 6.35e-03 | 6.35e-03 | 7.51e-06 | 0.12 | 6.34e-03 | 6.36e-03 | SI | 0.00 |
| cross | 1,024 | 10 | 11 | 0.05 | 2.54e+02 | 2.82e-04 | 2.82e-04 | 3.57e-07 | 0.13 | 2.82e-04 | 2.83e-04 | SI | 0.01 |
| cube | 4 | 2 | 2 | 0.05 | 1.18e-01 | 4.00e+00 | 4.00e+00 | 3.92e-03 | 0.10 | 3.99e+00 | 4.01e+00 | SI | 0.00 |
| cube | 6 | 3 | 3 | 0.05 | 4.16e-01 | 8.00e+00 | 8.00e+00 | 1.25e-02 | 0.16 | 7.97e+00 | 8.02e+00 | SI | 0.01 |
| cube | 8 | 4 | 4 | 0.05 | 9.70e-01 | 1.60e+01 | 1.60e+01 | 2.25e-02 | 0.14 | 1.60e+01 | 1.61e+01 | SI | 0.06 |
| cube | 10 | 5 | 6 | 0.05 | 2.20e+00 | 3.20e+01 | 3.20e+01 | 4.81e-02 | 0.15 | 3.19e+01 | 3.21e+01 | SI | 0.01 |
| cube | 12 | 6 | 7 | 0.05 | 3.95e+00 | 6.40e+01 | 6.40e+01 | 9.99e-02 | 0.16 | 6.38e+01 | 6.42e+01 | SI | 0.01 |
| cube | 14 | 7 | 8 | 0.05 | 7.00e+00 | 1.28e+02 | 1.28e+02 | 2.10e-01 | 0.16 | 1.28e+02 | 1.29e+02 | SI | 0.02 |
| cube | 16 | 8 | 9 | 0.05 | 1.08e+01 | 2.64e+02 | 2.56e+02 | 3.27e-01 | 0.13 | 2.55e+02 | 2.57e+02 | SI | 3.03 |
| cube | 18 | 9 | 10 | 0.05 | 1.68e+01 | 5.12e+02 | 5.12e+02 | 7.31e-01 | 0.14 | 5.11e+02 | 5.13e+02 | SI | 0.02 |
| cube | 20 | 10 | 11 | 0.05 | 2.38e+01 | 1.02e+03 | 1.02e+03 | 1.98e+00 | 0.19 | 1.02e+03 | 1.03e+03 | SI | 0.00 |
| cube | 22 | 11 | 12 | 0.05 | 3.35e+01 | 2.05e+03 | 2.05e+03 | 3.28e+00 | 0.16 | 2.04e+03 | 2.05e+03 | SI | 0.01 |
| cube | 24 | 12 | 13 | 0.05 | 4.51e+01 | 4.10e+03 | 4.09e+03 | 8.18e+00 | 0.20 | 4.08e+03 | 4.11e+03 | SI | 0.07 |
| cube | 26 | 13 | 14 | 0.05 | 6.50e+01 | 8.19e+03 | 8.38e+03 | 8.09e+02 | 9.66 | 1.19e+04 | 8.23e+03 | NO | 2.27 |
| cube | 28 | 14 | 15 | 0.05 | 8.19e+01 | 1.64e+04 | 1.64e+04 | 2.58e+01 | 0.16 | 1.63e+04 | 1.64e+04 | SI | 0.02 |
| rh_10_20 | 20 | 10 | 11 | 0.05 | 5.22e+01 | 1.39e+04 | 1.39e+04 | 4.33e+01 | 0.31 | 1.38e+04 | 1.39e+04 | SI | 0.06 |
| rh_10_25 | 25 | 10 | 11 | 0.05 | 4.86e+01 | 5.73e+03 | 5.73e+03 | 9.95e+00 | 0.17 | 5.71e+03 | 5.75e+03 | SI | 0.03 |
| rh_10_30 | 30 | 10 | 11 | 0.05 | 4.56e+01 | 2.02e+03 | 2.02e+03 | 5.00e+00 | 0.25 | 2.01e+03 | 2.02e+03 | SI | 0.03 |
| rh_8_20 | 20 | 8 | 9 | 0.05 | 3.95e+01 | 3.76e+04 | 3.77e+04 | 3.97e+02 | 1.05 | 3.69e+04 | 3.85e+04 | SI | 0.28 |
| rh_8_25 | 25 | 8 | 9 | 0.05 | 2.40e+01 | 7.86e+02 | 7.86e+02 | 1.66e+00 | 0.21 | 7.83e+02 | 7.89e+02 | SI | 0.06 |
| rh_8_30 | 30 | 8 | 9 | 0.05 | 1.73e+01 | 2.47e+02 | 2.47e+02 | 3.53e-01 | 0.14 | 2.47e+02 | 2.48e+02 | SI | 0.01 |

Tabla 5.1: Resultados SoB

| <i>P</i> | <i>m</i> | <i>d</i> | <i>W</i> | ϵ | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|------------|------------|----------|----------|----------|----------------|----------|----------|-------------------------|---------|
| birkhoff | 9 | 4 | 4 | 0.05 | 2.75e-02 | 1.12e+00 | 1.12e+00 | 9.46e-02 | 8.43 | 1.00e+00 | 1.43e+00 | SI | 0.28 |
| birkhoff | 16 | 9 | 10 | 0.05 | 1.58e-01 | 6.21e-02 | 6.23e-02 | 3.91e-03 | 6.28 | 5.42e-02 | 7.38e-02 | SI | 0.30 |
| birkhoff | 25 | 16 | 18 | 0.05 | 6.64e-01 | 1.41e-04 | 1.39e-04 | 6.09e-06 | 4.37 | 1.31e-04 | 1.53e-04 | SI | 1.05 |
| birkhoff | 36 | 25 | 28 | 0.05 | 2.41e+00 | 7.35e-09 | 7.40e-09 | 3.12e-10 | 4.22 | 6.81e-09 | 8.07e-09 | SI | 0.70 |
| birkhoff | 49 | 36 | 40 | 0.05 | 8.01e+00 | 5.64e-15 | 5.56e-15 | 2.75e-16 | 4.95 | 5.06e-15 | 6.01e-15 | SI | 1.49 |
| birkhoff | 64 | 49 | 54 | 0.05 | 2.91e+01 | 4.42e-23 | 4.42e-23 | 2.13e-24 | 4.83 | 4.10e-23 | 4.92e-23 | SI | 0.04 |
| birkhoff | 81 | 64 | 70 | 0.05 | 6.75e+01 | 2.60e-33 | 2.57e-33 | 1.30e-34 | 5.05 | 2.28e-33 | 2.79e-33 | SI | 1.24 |
| birkhoff | 100 | 81 | 89 | 0.05 | 1.55e+02 | 8.78e-46 | 8.81e-46 | 3.85e-47 | 4.37 | 8.30e-46 | 9.65e-46 | SI | 0.33 |
| cc_8_10 | 70 | 8 | 1000 | 0.70 | 7.26e+00 | 1.57e+05 | 2.62e+06 | 3.07e+05 | 11.74 | 2.17e+06 | 3.24e+06 | SI | 1570.11 |
| cc_8_10 | 70 | 8 | 10000 | 0.70 | 1.70e+02 | 1.57e+05 | 2.67e+06 | 3.42e+05 | 12.78 | 2.10e+06 | 3.29e+06 | SI | 1605.50 |
| cc_8_10 | 70 | 8 | 9 | 0.05 | 4.09e-01 | 1.57e+05 | 1.95e+06 | 6.95e+05 | 35.60 | 1.05e+06 | 9.90e+05 | NO | 1144.06 |
| cc_8_11 | 88 | 8 | 1000 | 0.70 | 1.20e+01 | 1.39e+06 | 1.11e+08 | 2.30e+07 | 20.85 | 1.03e+08 | 9.11e+07 | NO | 7840.38 |
| cc_8_11 | 88 | 8 | 10000 | 0.70 | 1.25e+02 | 1.39e+06 | 1.22e+08 | 2.29e+07 | 18.79 | 1.00e+08 | 9.74e+07 | NO | 8667.00 |
| cc_8_11 | 88 | 8 | 9 | 0.05 | 5.73e-01 | 1.39e+06 | 3.35e+07 | 3.16e+07 | 94.42 | 1.60e+07 | 4.70e+07 | SI | 2308.23 |
| cc_8_5 | 10 | 8 | 9 | 0.05 | 2.40e-01 | 1.74e-03 | 1.63e-03 | 3.91e-04 | 24.01 | 1.04e-03 | 2.92e-03 | SI | 6.17 |
| cc_8_5 | 10 | 8 | 1000 | 0.70 | 3.37e+00 | 1.74e-03 | 1.66e-03 | 1.42e-04 | 8.58 | 1.43e-03 | 1.95e-03 | SI | 4.47 |
| cc_8_5 | 10 | 8 | 10000 | 0.70 | 3.37e+01 | 1.74e-03 | 1.77e-03 | 1.85e-04 | 10.43 | 1.28e-03 | 2.04e-03 | SI | 2.01 |
| cc_8_6 | 18 | 8 | 10000 | 0.70 | 4.71e+01 | 4.44e-01 | 4.41e-01 | 5.39e-02 | 12.22 | 3.52e-01 | 5.60e-01 | SI | 0.67 |
| cc_8_6 | 18 | 8 | 9 | 0.05 | 2.69e-01 | 4.44e-01 | 3.57e-01 | 8.04e-02 | 22.51 | 2.45e-01 | 6.06e-01 | SI | 19.66 |
| cc_8_6 | 18 | 8 | 1000 | 0.70 | 3.82e+00 | 4.44e-01 | 4.23e-01 | 5.32e-02 | 12.58 | 3.44e-01 | 5.30e-01 | SI | 4.90 |
| cc_8_8 | 40 | 8 | 9 | 0.05 | 2.93e-01 | 7.84e+02 | 2.52e+03 | 7.32e+02 | 29.05 | 1.53e+03 | 4.84e+03 | SI | 221.22 |
| cc_8_8 | 40 | 8 | 10000 | 0.70 | 4.71e+01 | 7.84e+02 | 3.22e+03 | 3.94e+02 | 12.25 | 2.66e+03 | 4.02e+03 | SI | 310.08 |
| cc_8_8 | 40 | 8 | 1000 | 0.70 | 4.79e+00 | 7.84e+02 | 3.26e+03 | 4.94e+02 | 15.14 | 2.50e+03 | 4.31e+03 | SI | 315.69 |
| cc_8_9 | 54 | 8 | 10000 | 0.70 | 4.92e+01 | 1.33e+04 | 1.98e+05 | 1.96e+04 | 9.90 | 1.61e+05 | 2.50e+05 | SI | 1383.11 |
| cc_8_9 | 54 | 8 | 9 | 0.05 | 2.76e-01 | 1.33e+04 | 1.76e+05 | 2.96e+04 | 16.89 | 1.26e+05 | 2.49e+05 | SI | 1215.70 |
| cc_8_9 | 54 | 8 | 1000 | 0.70 | 4.78e+00 | 1.33e+04 | 2.15e+05 | 1.35e+04 | 6.27 | 1.88e+05 | 2.38e+05 | SI | 1511.66 |
| ccp | 56 | 10 | 11 | 0.05 | 1.38e-01 | 2.31e+00 | 2.34e+00 | 1.20e-01 | 5.13 | 2.13e+00 | 2.62e+00 | SI | 1.19 |
| ccp | 56 | 10 | 1000 | 0.05 | 9.49e+00 | 2.31e+00 | 2.33e+00 | 7.92e-02 | 3.40 | 2.13e+00 | 2.45e+00 | SI | 0.66 |
| cross | 4 | 2 | 2 | 0.05 | 6.13e-03 | 2.00e+00 | 2.02e+00 | 6.34e-02 | 3.14 | 1.91e+00 | 2.12e+00 | SI | 1.04 |
| cross | 8 | 3 | 3 | 0.05 | 9.69e-03 | 1.33e+00 | 1.34e+00 | 5.46e-02 | 4.09 | 1.24e+00 | 1.47e+00 | SI | 0.26 |
| cross | 16 | 4 | 4 | 0.05 | 1.73e-02 | 6.67e-01 | 6.63e-01 | 3.21e-02 | 4.84 | 5.92e-01 | 7.14e-01 | SI | 0.53 |
| cross | 32 | 5 | 6 | 0.05 | 3.35e-02 | 2.67e-01 | 2.62e-01 | 7.06e-03 | 2.70 | 2.49e-01 | 2.74e-01 | SI | 1.93 |
| cross | 64 | 6 | 7 | 0.05 | 6.48e-02 | 8.89e-02 | 8.86e-02 | 2.83e-03 | 3.19 | 8.52e-02 | 9.56e-02 | SI | 0.28 |
| cross | 128 | 7 | 8 | 0.05 | 1.40e-01 | 2.54e-02 | 2.52e-02 | 1.01e-03 | 4.02 | 2.24e-02 | 2.66e-02 | SI | 0.82 |
| cross | 256 | 8 | 9 | 0.05 | 2.92e-01 | 6.35e-03 | 6.38e-03 | 2.37e-04 | 3.71 | 5.82e-03 | 6.96e-03 | SI | 0.48 |
| cross | 512 | 9 | 10 | 0.05 | 6.37e-01 | 1.41e-03 | 1.39e-03 | 6.69e-05 | 4.81 | 1.26e-03 | 1.51e-03 | SI | 1.37 |
| cross | 1,024 | 10 | 11 | 0.05 | 1.54e+00 | 2.82e-04 | 2.87e-04 | 1.04e-05 | 3.64 | 2.66e-04 | 3.08e-04 | SI | 1.54 |
| cross | 2,048 | 11 | 12 | 0.05 | 3.37e+00 | 5.13e-05 | 5.09e-05 | 1.69e-06 | 3.32 | 4.78e-05 | 5.52e-05 | SI | 0.84 |
| cross | 4,096 | 12 | 13 | 0.05 | 8.26e+00 | 8.55e-06 | 8.46e-06 | 2.83e-07 | 3.34 | 7.92e-06 | 9.21e-06 | SI | 1.02 |
| cube | 4 | 2 | 1000 | 0.05 | 9.27e-01 | 4.00e+00 | 4.01e+00 | 1.23e-01 | 3.08 | 3.73e+00 | 4.22e+00 | SI | 0.19 |

| <i>P</i> | <i>m</i> | <i>d</i> | <i>W</i> | ϵ | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|------------|------------|----------|----------|----------|----------------|----------|----------|-------------------------|---------|
| cube | 6 | 3 | 3 | 0.05 | 1.02e-02 | 8.00e+00 | 8.11e+00 | 3.31e-01 | 4.08 | 7.28e+00 | 8.66e+00 | SI | 1.34 |
| cube | 6 | 3 | 1000 | 0.05 | 1.39e+00 | 8.00e+00 | 7.90e+00 | 1.78e-01 | 2.25 | 7.55e+00 | 8.18e+00 | SI | 1.22 |
| cube | 8 | 4 | 4 | 0.05 | 1.37e-02 | 1.60e+01 | 1.61e+01 | 4.24e-01 | 2.63 | 1.52e+01 | 1.72e+01 | SI | 0.86 |
| cube | 8 | 4 | 1000 | 0.05 | 1.53e+00 | 1.60e+01 | 1.59e+01 | 5.12e-01 | 3.23 | 1.49e+01 | 1.68e+01 | SI | 0.84 |
| cube | 10 | 5 | 1000 | 0.05 | 1.93e+00 | 3.20e+01 | 3.25e+01 | 1.30e+00 | 4.01 | 3.01e+01 | 3.59e+01 | SI | 1.66 |
| cube | 10 | 5 | 6 | 0.05 | 1.97e-02 | 3.20e+01 | 3.20e+01 | 9.18e-01 | 2.86 | 3.05e+01 | 3.40e+01 | SI | 0.08 |
| cube | 12 | 6 | 7 | 0.05 | 2.77e-02 | 6.40e+01 | 6.39e+01 | 2.71e+00 | 4.25 | 5.80e+01 | 6.84e+01 | SI | 0.22 |
| cube | 12 | 6 | 1000 | 0.05 | 2.18e+00 | 6.40e+01 | 6.41e+01 | 2.33e+00 | 3.63 | 5.80e+01 | 6.91e+01 | SI | 0.10 |
| cube | 14 | 7 | 8 | 0.05 | 4.13e-02 | 1.28e+02 | 1.28e+02 | 5.24e+00 | 4.10 | 1.15e+02 | 1.38e+02 | SI | 0.16 |
| cube | 14 | 7 | 1000 | 0.05 | 2.66e+00 | 1.28e+02 | 1.29e+02 | 4.05e+00 | 3.15 | 1.20e+02 | 1.35e+02 | SI | 0.55 |
| cube | 16 | 8 | 9 | 0.05 | 4.19e-02 | 2.64e+02 | 2.55e+02 | 9.76e+00 | 3.83 | 2.36e+02 | 2.75e+02 | SI | 3.59 |
| cube | 16 | 8 | 1000 | 0.05 | 2.96e+00 | 2.64e+02 | 2.57e+02 | 7.44e+00 | 2.90 | 2.44e+02 | 2.69e+02 | SI | 2.75 |
| cube | 18 | 9 | 1000 | 0.05 | 3.19e+00 | 5.12e+02 | 5.10e+02 | 1.77e+01 | 3.47 | 4.82e+02 | 5.49e+02 | SI | 0.44 |
| cube | 18 | 9 | 10 | 0.05 | 5.07e-02 | 5.12e+02 | 5.11e+02 | 1.72e+01 | 3.37 | 4.87e+02 | 5.68e+02 | SI | 0.26 |
| cube | 20 | 10 | 11 | 0.05 | 5.85e-02 | 1.02e+03 | 1.03e+03 | 5.09e+01 | 4.92 | 1.01e+03 | 9.99e+02 | NO | 1.01 |
| cube | 20 | 10 | 1000 | 0.05 | 3.47e+00 | 1.02e+03 | 1.01e+03 | 3.30e+01 | 3.25 | 1.00e+03 | 9.98e+02 | NO | 0.89 |
| cube | 22 | 11 | 1000 | 0.05 | 3.79e+00 | 2.05e+03 | 2.07e+03 | 6.18e+01 | 2.99 | 1.95e+03 | 2.19e+03 | SI | 0.90 |
| cube | 22 | 11 | 12 | 0.05 | 6.81e-02 | 2.05e+03 | 2.03e+03 | 9.56e+01 | 4.71 | 1.90e+03 | 2.20e+03 | SI | 1.01 |
| cube | 24 | 12 | 1000 | 0.05 | 4.29e+00 | 4.10e+03 | 4.10e+03 | 1.26e+02 | 3.07 | 3.87e+03 | 4.37e+03 | SI | 0.22 |
| cube | 24 | 12 | 13 | 0.05 | 8.64e-02 | 4.10e+03 | 4.10e+03 | 1.42e+02 | 3.45 | 3.82e+03 | 4.36e+03 | SI | 0.15 |
| cube | 26 | 13 | 14 | 0.05 | 9.56e-02 | 8.19e+03 | 8.04e+03 | 2.81e+02 | 3.50 | 7.50e+03 | 8.58e+03 | SI | 1.89 |
| cube | 26 | 13 | 1000 | 0.05 | 5.03e+00 | 8.19e+03 | 8.16e+03 | 2.55e+02 | 3.13 | 7.77e+03 | 8.70e+03 | SI | 0.39 |
| cube | 28 | 14 | 15 | 0.05 | 1.19e-01 | 1.64e+04 | 1.66e+04 | 7.19e+02 | 4.32 | 1.55e+04 | 1.77e+04 | SI | 1.55 |
| cube | 28 | 14 | 1000 | 0.05 | 5.56e+00 | 1.64e+04 | 1.67e+04 | 5.56e+02 | 3.32 | 1.60e+04 | 1.82e+04 | SI | 2.05 |
| metric | 6 | 6 | 7 | 0.05 | 8.72e-02 | 8.77e+01 | 8.59e+01 | 7.24e+00 | 8.44 | 1.08e+02 | 9.66e+01 | NO | 2.12 |
| metric | 25 | 10 | 11 | 0.05 | 1.55e-01 | 7.11e+03 | 7.06e+03 | 3.03e+02 | 4.29 | 6.51e+03 | 7.55e+03 | SI | 0.68 |
| rh_10_20 | 20 | 10 | 11 | 0.05 | 1.45e-01 | 1.39e+04 | 1.35e+04 | 9.43e+02 | 6.97 | 1.19e+04 | 1.60e+04 | SI | 2.52 |
| rh_10_20 | 20 | 10 | 1000 | 0.05 | 9.20e+00 | 1.39e+04 | 1.39e+04 | 6.16e+02 | 4.44 | 1.28e+04 | 1.51e+04 | SI | 0.02 |
| rh_10_25 | 25 | 10 | 1000 | 0.05 | 9.50e+00 | 5.73e+03 | 5.72e+03 | 2.18e+02 | 3.82 | 5.31e+03 | 6.10e+03 | SI | 0.20 |
| rh_10_25 | 25 | 10 | 11 | 0.05 | 1.41e-01 | 5.73e+03 | 5.77e+03 | 3.28e+02 | 5.68 | 5.30e+03 | 6.71e+03 | SI | 0.79 |
| rh_10_30 | 30 | 10 | 1000 | 0.05 | 8.82e+00 | 2.02e+03 | 2.02e+03 | 4.78e+01 | 2.37 | 1.94e+03 | 2.13e+03 | SI | 0.17 |
| rh_10_30 | 30 | 10 | 11 | 0.05 | 1.29e-01 | 2.02e+03 | 2.03e+03 | 7.78e+01 | 3.84 | 1.87e+03 | 2.15e+03 | SI | 0.60 |
| rh_8_20 | 20 | 8 | 1000 | 0.05 | 1.67e+01 | 3.76e+04 | 3.67e+04 | 2.62e+03 | 7.15 | 3.26e+04 | 4.09e+04 | SI | 2.43 |
| rh_8_20 | 20 | 8 | 9 | 0.05 | 2.17e-01 | 3.76e+04 | 2.78e+04 | 7.09e+03 | 25.47 | 1.47e+04 | 4.30e+04 | SI | 25.89 |
| rh_8_25 | 25 | 8 | 9 | 0.05 | 1.00e-01 | 7.86e+02 | 7.66e+02 | 4.89e+01 | 6.38 | 6.91e+02 | 9.18e+02 | SI | 2.51 |
| rh_8_25 | 25 | 8 | 1000 | 0.05 | 8.54e+00 | 7.86e+02 | 7.92e+02 | 2.86e+01 | 3.61 | 7.45e+02 | 8.67e+02 | SI | 0.77 |
| rh_8_30 | 30 | 8 | 9 | 0.05 | 7.91e-02 | 2.47e+02 | 2.49e+02 | 6.68e+00 | 2.68 | 2.38e+02 | 2.61e+02 | SI | 0.81 |
| rh_8_30 | 30 | 8 | 1000 | 0.05 | 6.44e+00 | 2.47e+02 | 2.46e+02 | 9.77e+00 | 3.97 | 2.30e+02 | 2.67e+02 | SI | 0.52 |

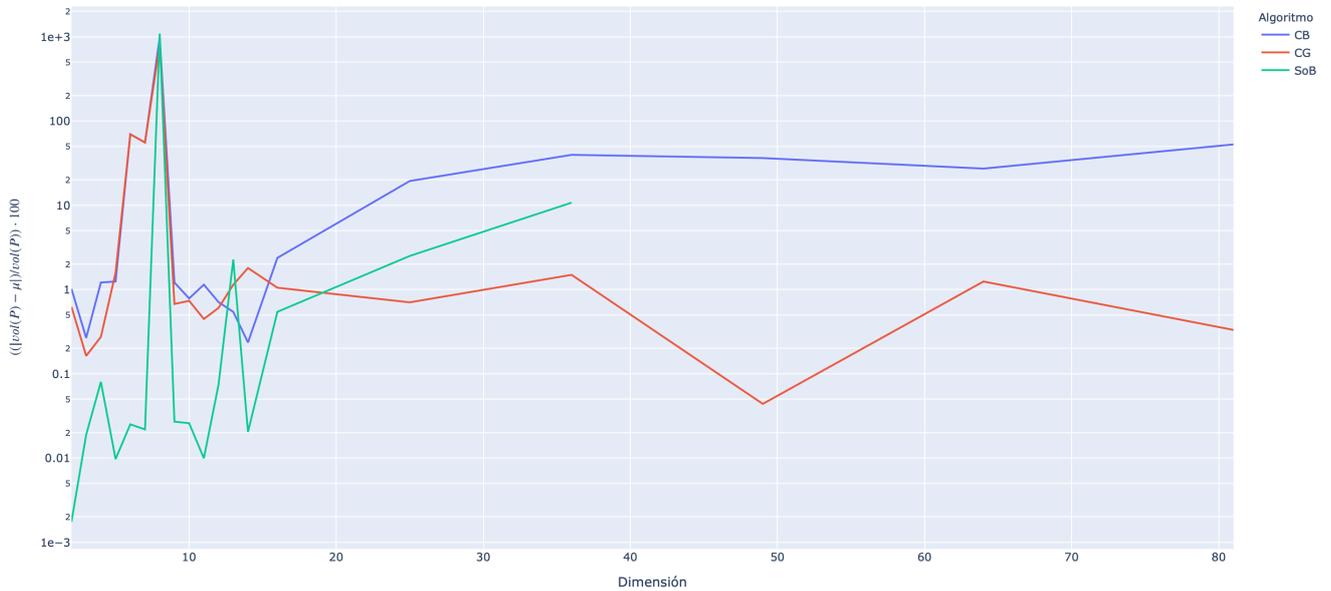
Tabla 5.2: Resultados CG

| <i>P</i> | <i>m</i> | <i>d</i> | <i>W</i> | ϵ | Tiempo (s) | Volumen | μ | σ | σ/μ % | <i>min</i> | <i>max</i> | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|------------|------------|----------|----------|----------|----------------|------------|------------|-------------------------|---------|
| birkhoff | 9 | 4 | 4 | 0.05 | 7.65e-03 | 1.12e+00 | 1.14e+00 | 6.47e-02 | 5.67 | 1.03e+00 | 1.32e+00 | SI | 1.42 |
| birkhoff | 9 | 4 | 10000 | 0.70 | 2.96e+00 | 1.12e+00 | 1.18e+00 | 1.11e-01 | 9.41 | 1.05e+00 | 9.92e-01 | NO | 5.28 |
| birkhoff | 16 | 9 | 10 | 0.05 | 4.04e-02 | 6.21e-02 | 6.04e-02 | 7.07e-03 | 11.70 | 4.67e-02 | 7.38e-02 | SI | 2.74 |
| birkhoff | 16 | 9 | 10000 | 0.70 | 9.30e+00 | 6.21e-02 | 6.13e-02 | 4.76e-03 | 7.77 | 5.32e-02 | 7.73e-02 | SI | 1.22 |
| birkhoff | 25 | 16 | 18 | 0.05 | 4.52e-01 | 1.41e-04 | 1.26e-04 | 1.82e-05 | 14.45 | 1.06e-04 | 9.65e-05 | NO | 10.50 |
| birkhoff | 25 | 16 | 10000 | 0.70 | 4.52e+01 | 1.41e-04 | 1.49e-04 | 1.87e-05 | 12.56 | 1.16e-04 | 1.95e-04 | SI | 5.76 |
| birkhoff | 36 | 25 | 28 | 0.05 | 3.18e+00 | 7.35e-09 | 4.36e-09 | 9.75e-10 | 22.37 | 2.91e-09 | 6.04e-09 | SI | 40.73 |
| birkhoff | 36 | 25 | 10000 | 0.70 | 1.41e+02 | 7.35e-09 | 7.49e-09 | 1.69e-09 | 22.52 | 1.00e-08 | 8.43e-09 | NO | 1.94 |
| birkhoff | 49 | 36 | 40 | 0.05 | 1.69e+01 | 5.64e-15 | 1.27e-15 | 6.32e-16 | 49.70 | 1.10e-15 | 8.68e-16 | NO | 77.47 |
| birkhoff | 49 | 36 | 10000 | 0.70 | 4.48e+02 | 5.64e-15 | 5.54e-15 | 1.73e-15 | 31.23 | 1.03e-14 | 8.48e-15 | NO | 1.80 |
| birkhoff | 64 | 49 | 54 | 0.05 | 5.66e+01 | 4.42e-23 | 1.05e-25 | 2.28e-25 | 217.35 | 1.24e-25 | 9.32e-27 | NO | 99.76 |
| birkhoff | 64 | 49 | 10000 | 0.70 | 2.34e+03 | 4.42e-23 | 4.62e-23 | 1.68e-23 | 36.37 | 2.22e-23 | 8.48e-23 | SI | 4.65 |
| birkhoff | 81 | 64 | 10000 | 0.70 | 2.52e+03 | 2.60e-33 | 1.89e-33 | 7.59e-34 | 40.13 | 1.13e-33 | 7.32e-34 | NO | 27.22 |
| birkhoff | 100 | 81 | 10000 | 0.70 | 9.69e+03 | 8.78e-46 | 4.16e-46 | 1.90e-46 | 45.81 | 1.41e-46 | 9.24e-47 | NO | 52.67 |
| cc_8_10 | 70 | 8 | 9 | 0.05 | 1.17e-01 | 1.57e+05 | 1.87e+06 | 6.46e+05 | 34.50 | 1.22e+06 | 9.77e+05 | NO | 1094.11 |
| cc_8_10 | 70 | 8 | 1000 | 0.70 | 2.32e+00 | 1.57e+05 | 2.39e+06 | 3.73e+05 | 15.59 | 1.69e+06 | 3.32e+06 | SI | 1425.67 |
| cc_8_10 | 70 | 8 | 10000 | 0.70 | 5.09e+01 | 1.57e+05 | 2.40e+06 | 3.08e+05 | 12.85 | 1.93e+06 | 3.22e+06 | SI | 1429.82 |
| cc_8_11 | 88 | 8 | 9 | 0.05 | 1.77e-01 | 1.39e+06 | 7.55e+07 | 6.36e+07 | 84.21 | 1.09e+07 | 8.48e+07 | SI | 5323.09 |
| cc_8_11 | 88 | 8 | 1000 | 0.70 | 3.84e+00 | 1.39e+06 | 1.32e+08 | 4.52e+07 | 34.20 | 1.05e+08 | 9.88e+07 | NO | 9403.81 |
| cc_8_11 | 88 | 8 | 10000 | 0.70 | 3.35e+01 | 1.39e+06 | 1.25e+08 | 1.61e+07 | 12.81 | 1.00e+08 | 9.38e+07 | NO | 8912.72 |
| cc_8_5 | 10 | 8 | 9 | 0.05 | 8.82e-02 | 1.74e-03 | 1.54e-03 | 7.62e-04 | 49.53 | 3.81e-04 | 3.27e-03 | SI | 11.44 |
| cc_8_5 | 10 | 8 | 1000 | 0.70 | 1.78e+00 | 1.74e-03 | 1.69e-03 | 2.91e-04 | 17.18 | 1.27e-03 | 2.40e-03 | SI | 2.48 |
| cc_8_5 | 10 | 8 | 10000 | 0.70 | 1.67e+01 | 1.74e-03 | 1.71e-03 | 2.42e-04 | 14.12 | 1.38e-03 | 2.14e-03 | SI | 1.22 |
| cc_8_6 | 18 | 8 | 1000 | 0.70 | 1.56e+00 | 4.44e-01 | 4.42e-01 | 5.86e-02 | 13.24 | 3.57e-01 | 6.14e-01 | SI | 0.45 |
| cc_8_6 | 18 | 8 | 10000 | 0.70 | 1.52e+01 | 4.44e-01 | 4.43e-01 | 5.69e-02 | 12.84 | 3.21e-01 | 5.46e-01 | SI | 0.37 |
| cc_8_6 | 18 | 8 | 9 | 0.05 | 7.99e-02 | 4.44e-01 | 3.88e-01 | 1.47e-01 | 37.87 | 2.08e-01 | 8.56e-01 | SI | 12.80 |
| cc_8_8 | 40 | 8 | 1000 | 0.70 | 1.71e+00 | 7.84e+02 | 3.08e+03 | 5.89e+02 | 19.11 | 2.21e+03 | 4.36e+03 | SI | 292.92 |
| cc_8_8 | 40 | 8 | 10000 | 0.70 | 1.67e+01 | 7.84e+02 | 3.09e+03 | 4.43e+02 | 14.33 | 2.21e+03 | 4.17e+03 | SI | 293.88 |
| cc_8_8 | 40 | 8 | 9 | 0.05 | 6.88e-02 | 7.84e+02 | 2.69e+03 | 8.41e+02 | 31.22 | 1.26e+03 | 3.96e+03 | SI | 243.60 |
| cc_8_9 | 54 | 8 | 9 | 0.05 | 4.99e-02 | 1.33e+04 | 1.97e+05 | 4.39e+04 | 22.27 | 1.25e+05 | 3.05e+05 | SI | 1379.08 |
| cc_8_9 | 54 | 8 | 1000 | 0.70 | 1.31e+00 | 1.33e+04 | 1.91e+05 | 2.32e+04 | 12.13 | 1.41e+05 | 2.35e+05 | SI | 1331.38 |
| cc_8_9 | 54 | 8 | 10000 | 0.70 | 1.29e+01 | 1.33e+04 | 2.03e+05 | 1.57e+04 | 7.74 | 1.76e+05 | 2.47e+05 | SI | 1420.96 |
| ccp | 56 | 10 | 11 | 0.05 | 1.76e-02 | 2.31e+00 | 2.34e+00 | 1.57e-01 | 6.71 | 2.05e+00 | 2.66e+00 | SI | 1.37 |
| ccp | 56 | 10 | 1000 | 0.05 | 6.83e-01 | 2.31e+00 | 2.27e+00 | 1.10e-01 | 4.86 | 2.12e+00 | 2.50e+00 | SI | 1.82 |
| cross | 4 | 2 | 2 | 0.05 | 5.60e-03 | 2.00e+00 | 2.03e+00 | 1.01e-01 | 4.98 | 1.81e+00 | 2.17e+00 | SI | 1.37 |
| cross | 8 | 3 | 3 | 0.05 | 6.14e-03 | 1.33e+00 | 1.33e+00 | 6.71e-02 | 5.03 | 1.25e+00 | 1.50e+00 | SI | 0.01 |
| cross | 16 | 4 | 4 | 0.05 | 6.75e-03 | 6.67e-01 | 6.67e-01 | 3.53e-02 | 5.29 | 6.11e-01 | 7.66e-01 | SI | 0.10 |
| cross | 32 | 5 | 6 | 0.05 | 8.48e-03 | 2.67e-01 | 2.65e-01 | 1.17e-02 | 4.42 | 2.45e-01 | 2.82e-01 | SI | 0.72 |
| cross | 64 | 6 | 7 | 0.05 | 1.23e-02 | 8.89e-02 | 8.78e-02 | 2.73e-03 | 3.12 | 8.33e-02 | 9.39e-02 | SI | 1.23 |
| cross | 128 | 7 | 8 | 0.05 | 1.53e-02 | 2.54e-02 | 2.52e-02 | 1.02e-03 | 4.07 | 2.30e-02 | 2.69e-02 | SI | 0.89 |
| cross | 256 | 8 | 9 | 0.05 | 2.20e-02 | 6.35e-03 | 6.37e-03 | 2.80e-04 | 4.40 | 5.89e-03 | 7.02e-03 | SI | 0.39 |
| cross | 512 | 9 | 10 | 0.05 | 3.40e-02 | 1.41e-03 | 1.40e-03 | 8.01e-05 | 5.73 | 1.25e-03 | 1.54e-03 | SI | 0.89 |

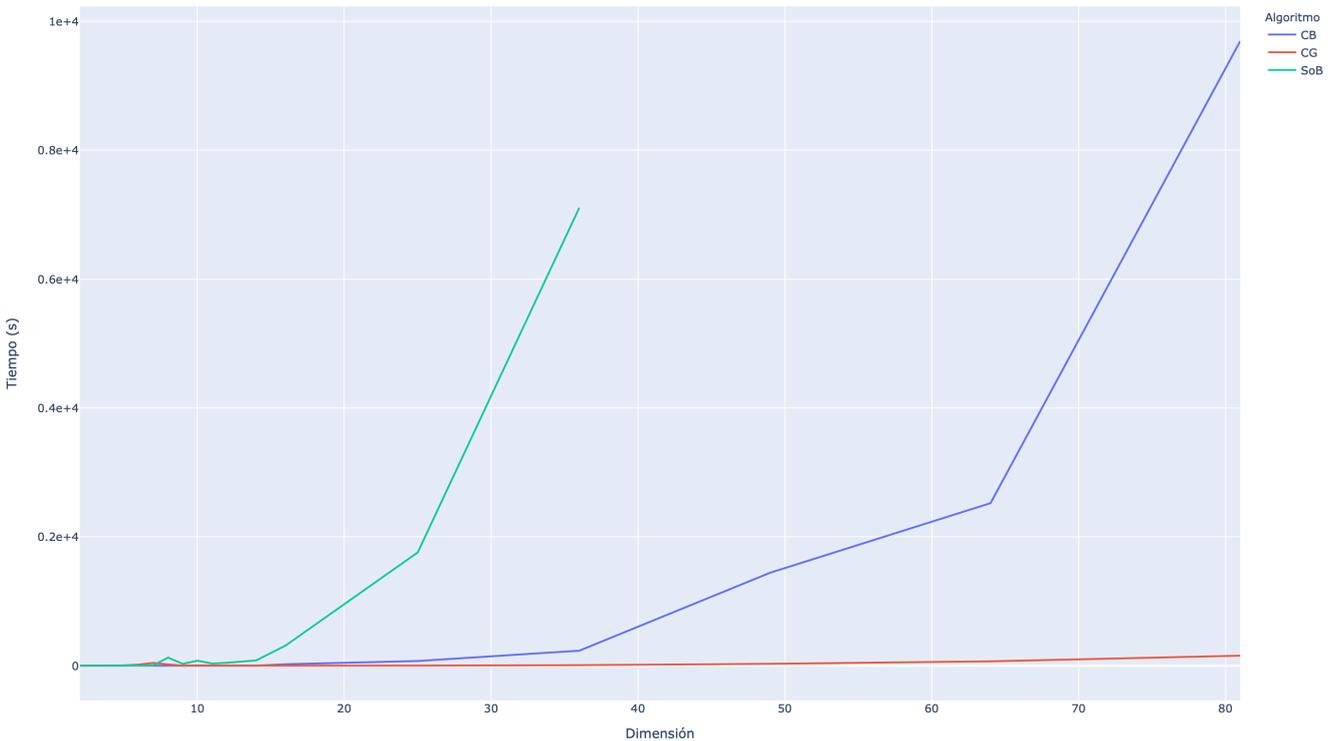
| <i>P</i> | <i>m</i> | <i>d</i> | <i>W</i> | ϵ | Tiempo (s) | Volumen | μ | σ | σ/μ % | <i>min</i> | <i>max</i> | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|------------|------------|----------|----------|----------|----------------|------------|------------|-------------------------|---------|
| cross | 2,048 | 11 | 12 | 0.05 | 1.86e-01 | 5.13e-05 | 5.07e-05 | 1.29e-06 | 2.55 | 4.85e-05 | 5.29e-05 | SI | 1.26 |
| cross | 4,096 | 12 | 13 | 0.05 | 3.18e-01 | 8.55e-06 | 8.48e-06 | 4.47e-07 | 5.27 | 7.69e-06 | 9.25e-06 | SI | 0.81 |
| cube | 4 | 2 | 1000 | 0.05 | 2.18e-01 | 4.00e+00 | 4.06e+00 | 2.40e-01 | 5.91 | 3.70e+00 | 4.52e+00 | SI | 1.43 |
| cube | 4 | 2 | 2 | 0.05 | 5.64e-03 | 4.00e+00 | 4.00e+00 | 1.75e-01 | 4.39 | 3.63e+00 | 4.36e+00 | SI | 0.10 |
| cube | 6 | 3 | 1000 | 0.05 | 2.46e-01 | 8.00e+00 | 7.99e+00 | 2.52e-01 | 3.15 | 7.49e+00 | 8.68e+00 | SI | 0.09 |
| cube | 6 | 3 | 3 | 0.05 | 6.27e-03 | 8.00e+00 | 7.92e+00 | 3.37e-01 | 4.25 | 7.47e+00 | 8.86e+00 | SI | 0.96 |
| cube | 8 | 4 | 1000 | 0.05 | 2.59e-01 | 1.60e+01 | 1.59e+01 | 8.47e-01 | 5.33 | 1.44e+01 | 1.72e+01 | SI | 0.59 |
| cube | 8 | 4 | 4 | 0.05 | 6.83e-03 | 1.60e+01 | 1.62e+01 | 6.96e-01 | 4.31 | 1.47e+01 | 1.71e+01 | SI | 0.95 |
| cube | 10 | 5 | 6 | 0.05 | 7.88e-03 | 3.20e+01 | 3.28e+01 | 1.19e+00 | 3.64 | 3.08e+01 | 3.58e+01 | SI | 2.49 |
| cube | 10 | 5 | 1000 | 0.05 | 2.91e-01 | 3.20e+01 | 3.23e+01 | 1.07e+00 | 3.30 | 3.01e+01 | 3.39e+01 | SI | 0.92 |
| cube | 12 | 6 | 1000 | 0.05 | 3.28e-01 | 6.40e+01 | 6.38e+01 | 3.09e+00 | 4.85 | 5.78e+01 | 7.23e+01 | SI | 0.35 |
| cube | 12 | 6 | 7 | 0.05 | 9.87e-03 | 6.40e+01 | 6.49e+01 | 3.87e+00 | 5.97 | 5.80e+01 | 7.40e+01 | SI | 1.36 |
| cube | 14 | 7 | 1000 | 0.05 | 4.11e-01 | 1.28e+02 | 1.27e+02 | 4.09e+00 | 3.23 | 1.17e+02 | 1.36e+02 | SI | 1.05 |
| cube | 14 | 7 | 8 | 0.05 | 1.18e-02 | 1.28e+02 | 1.27e+02 | 4.72e+00 | 3.72 | 1.18e+02 | 1.37e+02 | SI | 1.05 |
| cube | 16 | 8 | 1000 | 0.05 | 4.37e-01 | 2.64e+02 | 2.52e+02 | 1.06e+01 | 4.20 | 2.25e+02 | 2.71e+02 | SI | 4.66 |
| cube | 16 | 8 | 9 | 0.05 | 1.03e-02 | 2.64e+02 | 2.56e+02 | 1.21e+01 | 4.74 | 2.33e+02 | 2.82e+02 | SI | 3.20 |
| cube | 18 | 9 | 1000 | 0.05 | 4.73e-01 | 5.12e+02 | 5.20e+02 | 2.39e+01 | 4.60 | 4.82e+02 | 5.78e+02 | SI | 1.60 |
| cube | 18 | 9 | 10 | 0.05 | 1.39e-02 | 5.12e+02 | 5.12e+02 | 2.64e+01 | 5.16 | 4.72e+02 | 5.69e+02 | SI | 0.07 |
| cube | 20 | 10 | 1000 | 0.05 | 5.12e-01 | 1.02e+03 | 1.03e+03 | 3.35e+01 | 3.24 | 1.01e+03 | 9.96e+02 | NO | 0.78 |
| cube | 20 | 10 | 11 | 0.05 | 1.62e-02 | 1.02e+03 | 1.01e+03 | 4.07e+01 | 4.02 | 1.01e+03 | 1.00e+03 | NO | 1.19 |
| cube | 22 | 11 | 12 | 0.05 | 1.62e-02 | 2.05e+03 | 2.01e+03 | 8.51e+01 | 4.24 | 1.88e+03 | 2.15e+03 | SI | 2.09 |
| cube | 22 | 11 | 1000 | 0.05 | 5.56e-01 | 2.05e+03 | 2.05e+03 | 7.46e+01 | 3.64 | 1.90e+03 | 2.22e+03 | SI | 0.05 |
| cube | 24 | 12 | 1000 | 0.05 | 5.94e-01 | 4.10e+03 | 4.00e+03 | 1.21e+02 | 3.02 | 3.80e+03 | 4.30e+03 | SI | 2.42 |
| cube | 24 | 12 | 13 | 0.05 | 2.19e-02 | 4.10e+03 | 4.14e+03 | 2.18e+02 | 5.27 | 3.74e+03 | 4.67e+03 | SI | 1.20 |
| cube | 26 | 13 | 14 | 0.05 | 4.53e-02 | 8.19e+03 | 8.29e+03 | 4.41e+02 | 5.33 | 7.50e+03 | 9.12e+03 | SI | 1.16 |
| cube | 26 | 13 | 1000 | 0.05 | 6.45e-01 | 8.19e+03 | 8.19e+03 | 3.38e+02 | 4.13 | 7.53e+03 | 8.87e+03 | SI | 0.08 |
| cube | 28 | 14 | 1000 | 0.05 | 7.10e-01 | 1.64e+04 | 1.64e+04 | 6.85e+02 | 4.19 | 1.47e+04 | 1.76e+04 | SI | 0.12 |
| cube | 28 | 14 | 15 | 0.05 | 2.25e-02 | 1.64e+04 | 1.63e+04 | 7.62e+02 | 4.66 | 1.51e+04 | 1.79e+04 | SI | 0.35 |
| metric | 6 | 6 | 7 | 0.05 | 1.29e-02 | 8.77e+01 | 8.75e+01 | 1.24e+01 | 14.11 | 1.08e+02 | 9.84e+01 | NO | 0.23 |
| metric | 25 | 10 | 11 | 0.05 | 1.87e-02 | 7.11e+03 | 7.04e+03 | 7.15e+02 | 10.15 | 5.79e+03 | 8.87e+03 | SI | 0.91 |
| rh_10_20 | 20 | 10 | 1000 | 0.05 | 5.36e-01 | 1.39e+04 | 1.37e+04 | 6.24e+02 | 4.54 | 1.27e+04 | 1.50e+04 | SI | 1.14 |
| rh_10_20 | 20 | 10 | 11 | 0.05 | 1.78e-02 | 1.39e+04 | 1.35e+04 | 1.11e+03 | 8.27 | 1.18e+04 | 1.62e+04 | SI | 2.89 |
| rh_10_25 | 25 | 10 | 1000 | 0.05 | 5.38e-01 | 5.73e+03 | 5.86e+03 | 2.95e+02 | 5.04 | 5.36e+03 | 6.28e+03 | SI | 2.22 |
| rh_10_25 | 25 | 10 | 11 | 0.05 | 1.71e-02 | 5.73e+03 | 5.53e+03 | 3.77e+02 | 6.82 | 4.76e+03 | 6.20e+03 | SI | 3.51 |
| rh_10_30 | 30 | 10 | 11 | 0.05 | 1.63e-02 | 2.02e+03 | 2.01e+03 | 1.22e+02 | 6.08 | 1.80e+03 | 2.25e+03 | SI | 0.14 |
| rh_10_30 | 30 | 10 | 1000 | 0.05 | 5.40e-01 | 2.02e+03 | 2.00e+03 | 8.46e+01 | 4.24 | 1.88e+03 | 2.17e+03 | SI | 1.02 |
| rh_8_20 | 20 | 8 | 9 | 0.05 | 5.07e-02 | 3.76e+04 | 3.83e+04 | 1.57e+04 | 40.99 | 2.33e+04 | 8.94e+04 | SI | 1.84 |
| rh_8_20 | 20 | 8 | 1000 | 0.05 | 1.32e+00 | 3.76e+04 | 3.88e+04 | 4.24e+03 | 10.94 | 3.14e+04 | 4.89e+04 | SI | 3.23 |
| rh_8_25 | 25 | 8 | 1000 | 0.05 | 4.63e-01 | 7.86e+02 | 7.73e+02 | 3.28e+01 | 4.24 | 7.18e+02 | 8.38e+02 | SI | 1.69 |
| rh_8_25 | 25 | 8 | 9 | 0.05 | 1.36e-02 | | | | | | | | |

5.1.1. Análisis de resultados

En las figuras 5.1a y 5.1b se puede ver la evolución del error de estimación y el tiempo de ejecución en función de la dimensión del politopo respectivamente.



(a) Error en función de la dimensión.



(b) Tiempo de ejecución en función de la dimensión

Figura 5.1: Error y tiempo de ejecución en función de la dimensión. Los datos se presentan en escala logarítmica.

Con respecto al tiempo de ejecución, vemos como el algoritmo Sob comienza a crecer de manera exponencial en dimensiones relativamente bajas, por este motivo es que no se aplicó el algoritmo al conjunto completo de casos de prueba. Por el contrario, para los algoritmos CG y CB, se puede apreciar como el tiempo de ejecución crece de forma polinomial.

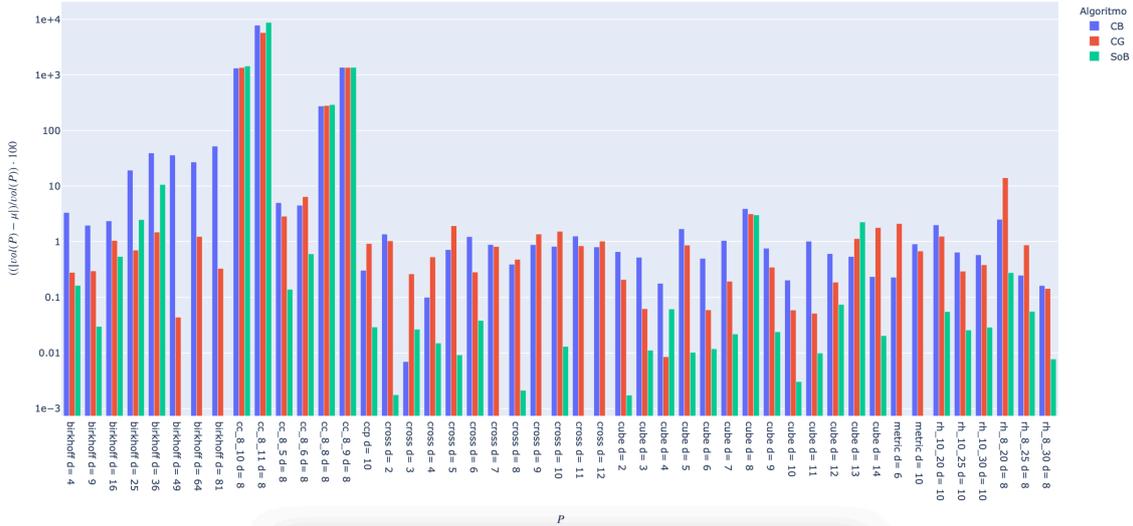


Figura 5.2: Error relativo porcentual calculado para cada politopo. Los datos se presentan en escala logarítmica.

Entrando más en detalle en el error, vemos que los politopos que tuvieron mayor error relativo, son los cc_m_n . El algoritmo SoB fue el que arrojó menor error relativo porcentual en la mayoría de los politopos (aunque cabe destacar que no fue posible aplicarlo en todos los politopos), seguido por el algoritmo CG.

Finalmente, como los algoritmos CG y CB fueron los que se pudieron aplicar a politopos de mayor dimensión, se decidió que estos algoritmos serán los que se van a aplicar al estudio de los modelos metabólicos.

5.2. Resultados sorteando muestras dentro del politopo

En este caso se realizaron las pruebas utilizando las cuatro caminatas aleatorias implementadas en la librería VolEsti (Random Walk, Billiard, CDHR, RDHR). En las tablas 5.4, 5.5, 5.6 y 5.7 se pueden ver dichos resultados. Las filas que se presentan en color verde son las filas de las estimaciones que cumplieron el criterio de aceptación que definimos (Error relativo porcentual menor al 10% y un coeficiente de variación menor a 25%).

| P | m | d | N | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $\min \leq \mu \leq \max$ | % Error |
|----------|-----|-----|----------|------------|----------|----------|----------|----------------|----------|----------|---------------------------|---------|
| birkhoff | 9 | 4 | 1.00e+05 | 3.22e+00 | 1.12e+00 | 1.12e+00 | 6.21e-03 | 0.55 | 1.11e+00 | 1.13e+00 | SI | 0.05 |
| birkhoff | 16 | 9 | 1.00e+05 | 6.78e+00 | 6.21e-02 | 6.21e-02 | 8.07e-04 | 1.30 | 6.05e-02 | 6.38e-02 | SI | 0.01 |
| birkhoff | 25 | 16 | 1.00e+05 | 1.39e+01 | 1.41e-04 | 1.44e-04 | 1.12e-05 | 7.78 | 1.33e-04 | 1.90e-04 | SI | 2.19 |
| birkhoff | 36 | 25 | 1.00e+05 | 2.36e+01 | 7.35e-09 | 7.29e-09 | 5.67e-10 | 7.79 | 6.56e-09 | 8.41e-09 | SI | 0.90 |
| birkhoff | 49 | 36 | 1.00e+05 | 4.26e+01 | 5.64e-15 | 5.51e-15 | 1.52e-15 | 27.66 | 3.18e-15 | 8.88e-15 | SI | 2.29 |
| cc_8_10 | 70 | 8 | 1.00e+05 | 4.46e+01 | 1.57e+05 | 2.45e+06 | 4.54e+04 | 1.86 | 2.34e+06 | 2.52e+06 | SI | 1460.39 |
| cc_8_11 | 88 | 8 | 1.00e+05 | 6.55e+01 | 1.39e+06 | 1.35e+08 | 1.44e+07 | 10.72 | 1.07e+08 | 1.56e+08 | SI | 9576.43 |
| cc_8_5 | 10 | 8 | 1.00e+05 | 4.03e+00 | 1.74e-03 | 1.75e-03 | 6.35e-05 | 3.62 | 1.59e-03 | 1.88e-03 | SI | 0.98 |
| cc_8_6 | 18 | 8 | 1.00e+05 | 7.54e+00 | 4.44e-01 | 4.44e-01 | 1.06e-02 | 2.39 | 4.29e-01 | 4.63e-01 | SI | 0.00 |
| cc_8_8 | 40 | 8 | 1.00e+05 | 2.01e+01 | 7.84e+02 | 3.06e+03 | 8.26e+01 | 2.70 | 2.86e+03 | 3.21e+03 | SI | 290.13 |
| cc_8_9 | 54 | 8 | 1.00e+05 | 3.37e+01 | 1.33e+04 | 1.97e+05 | 3.09e+03 | 1.56 | 1.92e+05 | 2.03e+05 | SI | 1380.38 |
| cross | 4 | 2 | 1.00e+05 | 1.19e+00 | 2.00e+00 | 2.00e+00 | 4.87e-03 | 0.24 | 1.99e+00 | 2.01e+00 | SI | 0.03 |
| cross | 8 | 3 | 1.00e+05 | 2.77e+00 | 1.33e+00 | 1.33e+00 | 6.69e-03 | 0.50 | 1.32e+00 | 1.34e+00 | SI | 0.13 |
| cross | 16 | 4 | 1.00e+05 | 6.53e+00 | 6.67e-01 | 6.67e-01 | 2.87e-03 | 0.43 | 6.62e-01 | 6.74e-01 | SI | 0.10 |
| cross | 32 | 5 | 1.00e+05 | 1.79e+01 | 2.67e-01 | 2.65e-01 | 9.54e-03 | 3.60 | 2.24e-01 | 2.70e-01 | SI | 0.73 |
| cross | 64 | 6 | 1.00e+05 | 5.29e+01 | 8.89e-02 | 8.88e-02 | 8.03e-04 | 0.90 | 8.74e-02 | 9.04e-02 | SI | 0.07 |
| cross | 128 | 7 | 1.00e+05 | 1.74e+02 | 2.54e-02 | 2.50e-02 | 1.57e-03 | 6.27 | 1.83e-02 | 2.61e-02 | SI | 1.59 |
| metric | 6 | 6 | 1.00e+05 | 2.95e+00 | 8.77e+01 | 8.67e+01 | 1.12e+00 | 1.29 | 8.48e+01 | 8.93e+01 | SI | 1.12 |
| metric | 25 | 10 | 1.00e+05 | 1.32e+01 | 7.11e+03 | 7.12e+03 | 2.14e+02 | 3.01 | 6.67e+03 | 7.51e+03 | SI | 0.10 |
| rh_10_20 | 20 | 10 | 1.00e+05 | 1.03e+01 | 1.39e+04 | 1.39e+04 | 4.05e+02 | 2.92 | 1.32e+04 | 1.51e+04 | SI | 0.02 |
| rh_10_25 | 25 | 10 | 1.00e+05 | 1.38e+01 | 5.73e+03 | 5.75e+03 | 1.16e+02 | 2.02 | 5.56e+03 | 6.00e+03 | SI | 0.34 |
| rh_10_30 | 30 | 10 | 1.00e+05 | 1.77e+01 | 2.02e+03 | 2.02e+03 | 2.32e+01 | 1.15 | 1.97e+03 | 2.08e+03 | SI | 0.07 |
| rh_8_20 | 20 | 8 | 1.00e+05 | 1.39e+01 | 3.76e+04 | 3.77e+04 | 1.31e+03 | 3.47 | 3.42e+04 | 3.98e+04 | SI | 0.42 |
| rh_8_25 | 25 | 8 | 1.00e+05 | 1.33e+01 | 7.86e+02 | 7.85e+02 | 1.06e+01 | 1.36 | 7.61e+02 | 8.07e+02 | SI | 0.10 |
| rh_8_30 | 30 | 8 | 1.00e+05 | 1.69e+01 | 2.47e+02 | 2.50e+02 | 1.13e+01 | 4.51 | 2.44e+02 | 2.98e+02 | SI | 1.00 |

Tabla 5.4: Resultados Ball Walk

| P | m | d | N | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $min \leq \mu \leq max$ | % Error |
|----------|-----|-----|----------|------------|----------|----------|----------|----------------|----------|----------|-------------------------|---------|
| birkhoff | 9 | 4 | 1.00e+05 | 3.21e+00 | 1.12e+00 | 1.12e+00 | 7.17e-03 | 0.64 | 1.11e+00 | 1.14e+00 | SI | 0.14 |
| birkhoff | 16 | 9 | 1.00e+05 | 6.86e+00 | 6.21e-02 | 6.23e-02 | 1.08e-03 | 1.73 | 5.96e-02 | 6.42e-02 | SI | 0.32 |
| birkhoff | 25 | 16 | 1.00e+05 | 1.30e+01 | 1.41e-04 | 1.41e-04 | 4.20e-06 | 2.97 | 1.34e-04 | 1.50e-04 | SI | 0.36 |
| birkhoff | 36 | 25 | 1.00e+05 | 2.36e+01 | 7.35e-09 | 7.19e-09 | 3.75e-10 | 5.22 | 6.49e-09 | 7.85e-09 | SI | 2.16 |
| birkhoff | 49 | 36 | 1.00e+05 | 4.28e+01 | 5.64e-15 | 5.70e-15 | 1.34e-15 | 23.57 | 3.39e-15 | 8.56e-15 | SI | 1.02 |
| cc_8_10 | 70 | 8 | 1.00e+05 | 4.52e+01 | 1.57e+05 | 2.45e+06 | 5.13e+04 | 2.09 | 2.33e+06 | 2.54e+06 | SI | 1461.36 |
| cc_8_11 | 88 | 8 | 1.00e+05 | 6.36e+01 | 1.39e+06 | 1.26e+08 | 1.40e+07 | 11.10 | 1.06e+08 | 1.64e+08 | SI | 8969.24 |
| cc_8_5 | 10 | 8 | 1.00e+05 | 3.96e+00 | 1.74e-03 | 1.73e-03 | 5.14e-05 | 2.96 | 1.62e-03 | 1.81e-03 | SI | 0.07 |
| cc_8_6 | 18 | 8 | 1.00e+05 | 7.53e+00 | 4.44e-01 | 4.25e-01 | 8.03e-02 | 18.88 | 7.77e-02 | 4.61e-01 | SI | 4.32 |
| cc_8_8 | 40 | 8 | 1.00e+05 | 2.01e+01 | 7.84e+02 | 3.12e+03 | 9.14e+01 | 2.93 | 2.93e+03 | 3.30e+03 | SI | 298.20 |
| cc_8_9 | 54 | 8 | 1.00e+05 | 3.30e+01 | 1.33e+04 | 1.96e+05 | 3.72e+03 | 1.90 | 1.90e+05 | 2.04e+05 | SI | 1370.06 |
| cross | 4 | 2 | 1.00e+05 | 1.19e+00 | 2.00e+00 | 2.00e+00 | 5.53e-03 | 0.28 | 1.99e+00 | 2.01e+00 | SI | 0.04 |
| cross | 8 | 3 | 1.00e+05 | 2.77e+00 | 1.33e+00 | 1.33e+00 | 4.66e-03 | 0.35 | 1.32e+00 | 1.34e+00 | SI | 0.03 |
| cross | 16 | 4 | 1.00e+05 | 6.50e+00 | 6.67e-01 | 6.66e-01 | 3.56e-03 | 0.53 | 6.59e-01 | 6.74e-01 | SI | 0.05 |
| cross | 32 | 5 | 1.00e+05 | 1.82e+01 | 2.67e-01 | 2.59e-01 | 3.54e-02 | 13.67 | 1.05e-01 | 2.69e-01 | SI | 3.00 |
| cross | 64 | 6 | 1.00e+05 | 5.48e+01 | 8.89e-02 | 8.87e-02 | 8.39e-04 | 0.95 | 8.59e-02 | 8.99e-02 | SI | 0.19 |
| cross | 128 | 7 | 1.00e+05 | 1.72e+02 | 2.54e-02 | 2.55e-02 | 2.58e-04 | 1.01 | 2.52e-02 | 2.63e-02 | SI | 0.41 |
| metric | 6 | 6 | 1.00e+05 | 3.02e+00 | 8.77e+01 | 8.62e+01 | 1.57e+00 | 1.82 | 8.39e+01 | 8.90e+01 | SI | 1.77 |
| metric | 25 | 10 | 1.00e+05 | 1.30e+01 | 7.11e+03 | 7.06e+03 | 2.34e+02 | 3.32 | 6.59e+03 | 7.52e+03 | SI | 0.72 |
| rh_10_20 | 20 | 10 | 1.00e+05 | 1.04e+01 | 1.39e+04 | 1.38e+04 | 2.68e+02 | 1.94 | 1.31e+04 | 1.43e+04 | SI | 0.36 |
| rh_10_25 | 25 | 10 | 1.00e+05 | 1.38e+01 | 5.73e+03 | 5.72e+03 | 9.80e+01 | 1.71 | 5.53e+03 | 5.87e+03 | SI | 0.11 |
| rh_10_30 | 30 | 10 | 1.00e+05 | 1.77e+01 | 2.02e+03 | 2.02e+03 | 3.15e+01 | 1.56 | 1.96e+03 | 2.08e+03 | SI | 0.15 |
| rh_8_20 | 20 | 8 | 1.00e+05 | 9.80e+00 | 3.76e+04 | 3.73e+04 | 1.73e+03 | 4.63 | 3.25e+04 | 4.01e+04 | SI | 0.65 |
| rh_8_25 | 25 | 8 | 1.00e+05 | 1.32e+01 | 7.86e+02 | 7.86e+02 | 6.03e+00 | 0.77 | 7.78e+02 | 8.00e+02 | SI | 0.05 |
| rh_8_30 | 30 | 8 | 1.00e+05 | 1.67e+01 | 2.47e+02 | 2.48e+02 | 3.27e+00 | 1.32 | 2.43e+02 | 2.54e+02 | SI | 0.06 |

Tabla 5.5: Resultados Billiard

| P | m | d | N | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $min \leq \mu \leq max$ | % Error |
|----------|-----|-----|----------|------------|----------|----------|----------|----------------|----------|----------|-------------------------|---------|
| birkhoff | 9 | 4 | 1.00e+05 | 3.20e+00 | 1.12e+00 | 1.13e+00 | 4.22e-03 | 0.37 | 1.12e+00 | 1.13e+00 | SI | 0.07 |
| birkhoff | 16 | 9 | 1.00e+05 | 6.87e+00 | 6.21e-02 | 6.19e-02 | 9.72e-04 | 1.57 | 6.01e-02 | 6.39e-02 | SI | 0.22 |
| birkhoff | 25 | 16 | 1.00e+05 | 1.29e+01 | 1.41e-04 | 1.39e-04 | 5.28e-06 | 3.80 | 1.29e-04 | 1.47e-04 | SI | 1.27 |
| birkhoff | 36 | 25 | 1.00e+05 | 2.36e+01 | 7.35e-09 | 7.42e-09 | 6.07e-10 | 8.18 | 5.99e-09 | 8.58e-09 | SI | 0.94 |
| birkhoff | 49 | 36 | 1.00e+05 | 4.25e+01 | 5.64e-15 | 5.52e-15 | 1.19e-15 | 21.59 | 3.48e-15 | 9.23e-15 | SI | 2.17 |
| birkhoff | 64 | 49 | 1.00e+05 | 7.18e+01 | 4.42e-23 | 5.00e-23 | 2.82e-23 | 56.42 | 1.12e-22 | 8.67e-23 | NO | 13.15 |
| cc_8_10 | 70 | 8 | 1.00e+05 | 4.90e+01 | 1.57e+05 | 2.45e+06 | 5.42e+04 | 2.21 | 2.37e+06 | 2.54e+06 | SI | 1463.62 |
| cc_8_11 | 88 | 8 | 1.00e+05 | 6.57e+01 | 1.39e+06 | 1.25e+08 | 1.25e+07 | 10.03 | 1.11e+08 | 9.74e+07 | NO | 8869.94 |
| cc_8_5 | 10 | 8 | 1.00e+05 | 3.96e+00 | 1.74e-03 | 1.73e-03 | 5.08e-05 | 2.94 | 1.63e-03 | 1.82e-03 | SI | 0.29 |
| cc_8_6 | 18 | 8 | 1.00e+05 | 7.53e+00 | 4.44e-01 | 4.45e-01 | 1.50e-02 | 3.36 | 4.26e-01 | 4.78e-01 | SI | 0.19 |
| cc_8_8 | 40 | 8 | 1.00e+05 | 2.01e+01 | 7.84e+02 | 3.11e+03 | 9.28e+01 | 2.98 | 2.90e+03 | 3.33e+03 | SI | 296.54 |
| cc_8_9 | 54 | 8 | 1.00e+05 | 3.38e+01 | 1.33e+04 | 1.98e+05 | 3.88e+03 | 1.96 | 1.93e+05 | 2.07e+05 | SI | 1385.59 |
| cross | 4 | 2 | 1.00e+05 | 1.08e+00 | 2.00e+00 | 2.00e+00 | 0.00e+00 | 0.00 | 2.00e+00 | 2.00e+00 | NO | 0.21 |
| cross | 8 | 3 | 1.00e+05 | 2.79e+00 | 1.33e+00 | 1.33e+00 | 5.49e-03 | 0.41 | 1.32e+00 | 1.34e+00 | SI | 0.04 |
| cross | 16 | 4 | 1.00e+05 | 6.45e+00 | 6.67e-01 | 6.67e-01 | 4.68e-03 | 0.70 | 6.58e-01 | 6.74e-01 | SI | 0.12 |
| cross | 32 | 5 | 1.00e+05 | 1.77e+01 | 2.67e-01 | 2.67e-01 | 2.07e-03 | 0.78 | 2.64e-01 | 2.72e-01 | SI | 0.14 |
| cross | 64 | 6 | 1.00e+05 | 5.24e+01 | 8.89e-02 | 8.93e-02 | 7.29e-04 | 0.82 | 8.77e-02 | 9.11e-02 | SI | 0.47 |
| cross | 128 | 7 | 1.00e+05 | 1.97e+02 | 2.54e-02 | 2.56e-02 | 8.24e-04 | 3.22 | 2.48e-02 | 2.93e-02 | SI | 0.83 |
| cross | 256 | 8 | 1.00e+05 | 6.17e+02 | 6.35e-03 | 6.35e-03 | 7.86e-05 | 1.24 | 6.19e-03 | 6.52e-03 | SI | 0.06 |
| metric | 6 | 6 | 1.00e+05 | 3.13e+00 | 8.77e+01 | 8.63e+01 | 1.51e+00 | 1.75 | 8.27e+01 | 8.88e+01 | SI | 1.66 |
| metric | 25 | 10 | 1.00e+05 | 1.33e+01 | 7.11e+03 | 7.09e+03 | 2.36e+02 | 3.33 | 6.72e+03 | 7.66e+03 | SI | 0.27 |
| rh_10_20 | 20 | 10 | 1.00e+05 | 1.03e+01 | 1.39e+04 | 1.41e+04 | 3.98e+02 | 2.83 | 1.30e+04 | 1.49e+04 | SI | 1.21 |
| rh_10_25 | 25 | 10 | 1.00e+05 | 1.38e+01 | 5.73e+03 | 5.71e+03 | 1.22e+02 | 2.13 | 5.50e+03 | 5.96e+03 | SI | 0.27 |
| rh_10_30 | 30 | 10 | 1.00e+05 | 1.76e+01 | 2.02e+03 | 2.01e+03 | 2.75e+01 | 1.37 | 1.95e+03 | 2.05e+03 | SI | 0.25 |
| rh_8_20 | 20 | 8 | 1.00e+05 | 1.01e+01 | 3.76e+04 | 3.81e+04 | 1.53e+03 | 4.02 | 3.58e+04 | 4.14e+04 | SI | 1.35 |
| rh_8_25 | 25 | 8 | 1.00e+05 | 1.32e+01 | 7.86e+02 | 7.89e+02 | 1.19e+01 | 1.51 | 7.57e+02 | 8.10e+02 | SI | 0.41 |
| rh_8_30 | 30 | 8 | 1.00e+05 | 1.68e+01 | 2.47e+02 | 2.47e+02 | 3.31e+00 | 1.34 | 2.41e+02 | 2.53e+02 | SI | 0.02 |

Tabla 5.6: Resultados CDHR

| P | m | d | N | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $\min \leq \mu \leq \max$ | % Error |
|----------|-----|-----|----------|------------|----------|----------|----------|----------------|----------|----------|---------------------------|---------|
| birkhoff | 9 | 4 | 1.00e+05 | 3.25e+00 | 1.12e+00 | 1.13e+00 | 6.35e-03 | 0.56 | 1.11e+00 | 1.14e+00 | SI | 0.20 |
| birkhoff | 16 | 9 | 1.00e+05 | 6.84e+00 | 6.21e-02 | 6.21e-02 | 7.12e-04 | 1.15 | 6.03e-02 | 6.31e-02 | SI | 0.03 |
| birkhoff | 25 | 16 | 1.00e+05 | 1.28e+01 | 1.41e-04 | 1.35e-04 | 1.97e-05 | 14.55 | 5.07e-05 | 1.44e-04 | SI | 4.10 |
| birkhoff | 36 | 25 | 1.00e+05 | 2.36e+01 | 7.35e-09 | 7.48e-09 | 5.42e-10 | 7.25 | 6.44e-09 | 8.75e-09 | SI | 1.72 |
| birkhoff | 49 | 36 | 1.00e+05 | 4.26e+01 | 5.64e-15 | 5.00e-15 | 8.58e-16 | 17.16 | 3.83e-15 | 6.98e-15 | SI | 11.35 |
| cc_8_10 | 70 | 8 | 1.00e+05 | 4.54e+01 | 1.57e+05 | 2.43e+06 | 7.51e+04 | 3.08 | 2.31e+06 | 2.58e+06 | SI | 1452.36 |
| cc_8_11 | 88 | 8 | 1.00e+05 | 6.77e+01 | 1.39e+06 | 1.26e+08 | 1.41e+07 | 11.24 | 1.05e+08 | 1.56e+08 | SI | 8928.16 |
| cc_8_5 | 10 | 8 | 1.00e+05 | 3.98e+00 | 1.74e-03 | 1.76e-03 | 5.21e-05 | 2.97 | 1.68e-03 | 1.88e-03 | SI | 1.21 |
| cc_8_6 | 18 | 8 | 1.00e+05 | 7.51e+00 | 4.44e-01 | 4.28e-01 | 9.06e-02 | 21.17 | 3.50e-02 | 4.63e-01 | SI | 3.68 |
| cc_8_8 | 40 | 8 | 1.00e+05 | 2.00e+01 | 7.84e+02 | 3.10e+03 | 8.39e+01 | 2.70 | 2.97e+03 | 3.31e+03 | SI | 295.63 |
| cc_8_9 | 54 | 8 | 1.00e+05 | 3.31e+01 | 1.33e+04 | 1.97e+05 | 4.38e+03 | 2.22 | 1.91e+05 | 2.09e+05 | SI | 1379.96 |
| cross | 4 | 2 | 1.00e+05 | 5.85e+01 | 2.00e+00 | 1.99e+00 | 4.28e-03 | 0.21 | 1.98e+00 | 2.00e+00 | SI | 0.31 |
| cross | 8 | 3 | 1.00e+05 | 2.79e+00 | 1.33e+00 | 1.33e+00 | 5.52e-03 | 0.41 | 1.32e+00 | 1.34e+00 | SI | 0.07 |
| cross | 16 | 4 | 1.00e+05 | 6.55e+00 | 6.67e-01 | 6.66e-01 | 4.10e-03 | 0.62 | 6.58e-01 | 6.76e-01 | SI | 0.11 |
| cross | 32 | 5 | 1.00e+05 | 1.76e+01 | 2.67e-01 | 2.67e-01 | 1.95e-03 | 0.73 | 2.63e-01 | 2.70e-01 | SI | 0.06 |
| cross | 64 | 6 | 1.00e+05 | 5.54e+01 | 8.89e-02 | 8.87e-02 | 7.98e-04 | 0.90 | 8.75e-02 | 9.07e-02 | SI | 0.24 |
| cross | 128 | 7 | 1.00e+05 | 1.91e+02 | 2.54e-02 | 2.52e-02 | 7.79e-04 | 3.09 | 2.21e-02 | 2.60e-02 | SI | 0.58 |
| metric | 6 | 6 | 1.00e+05 | 2.89e+00 | 8.77e+01 | 8.63e+01 | 1.05e+00 | 1.21 | 8.47e+01 | 8.80e+01 | SI | 1.60 |
| metric | 25 | 10 | 1.00e+05 | 1.32e+01 | 7.11e+03 | 7.10e+03 | 1.69e+02 | 2.38 | 6.82e+03 | 7.45e+03 | SI | 0.18 |
| rh_10_20 | 20 | 10 | 1.00e+05 | 1.04e+01 | 1.39e+04 | 1.39e+04 | 3.87e+02 | 2.78 | 1.33e+04 | 1.46e+04 | SI | 0.26 |
| rh_10_25 | 25 | 10 | 1.00e+05 | 1.39e+01 | 5.73e+03 | 5.74e+03 | 1.30e+02 | 2.27 | 5.58e+03 | 6.01e+03 | SI | 0.27 |
| rh_10_30 | 30 | 10 | 1.00e+05 | 1.76e+01 | 2.02e+03 | 1.97e+03 | 1.77e+02 | 8.94 | 1.22e+03 | 2.07e+03 | SI | 2.02 |
| rh_8_20 | 20 | 8 | 1.00e+05 | 9.81e+00 | 3.76e+04 | 3.79e+04 | 1.99e+03 | 5.25 | 3.42e+04 | 4.25e+04 | SI | 0.98 |
| rh_8_25 | 25 | 8 | 1.00e+05 | 1.33e+01 | 7.86e+02 | 7.87e+02 | 8.06e+00 | 1.02 | 7.71e+02 | 8.01e+02 | SI | 0.09 |
| rh_8_30 | 30 | 8 | 1.00e+05 | 1.69e+01 | 2.47e+02 | 2.48e+02 | 2.88e+00 | 1.16 | 2.41e+02 | 2.54e+02 | SI | 0.24 |

Tabla 5.7: Resultados RDHR

5.2.1. Análisis de resultados

En las figuras 5.8 y 5.4a se puede ver la evolución del error de estimación y el tiempo de ejecución en función de la dimensión respectivamente. Los datos se presentan discriminados por caminata aleatoria.

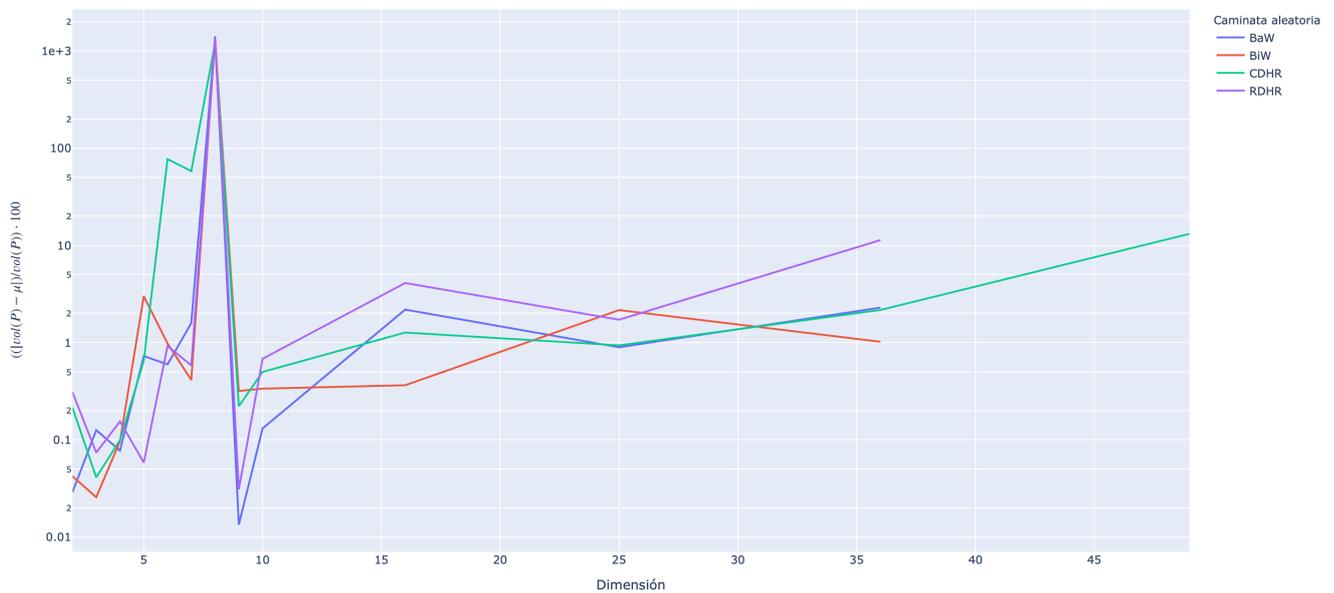


Tabla 5.8: Error en función de la dimensión. Los datos se presentan en escala logarítmica.

Con respecto al error, vemos que en general hay un promedio de error relativo bajo. Pero en el caso de los polítopos que hacen que se presente el pico que se ve en la gráfica 5.8, se puede observar en la tabla 5.9 y en la figura 5.3 que son polítopos que tienen un volumen muy pequeño, en relación al espacio de referencia y al mismo tiempo, este espacio es de volumen muy grande. Esto era de esperarse debido a la importancia que tiene para el método de Monte Carlo Directo la selección de un espacio de referencia que se ajuste lo más posible al polítopo.

| P | Volumen / Volumen ER | Volumen ER | % Error |
|----------------|----------------------|------------|----------|
| cross d= 6 | 1.39e-03 | 6.40e+01 | 0.01 |
| cross d= 4 | 4.17e-02 | 1.60e+01 | 0.02 |
| birkhoff d= 4 | 1.00e+00 | 1.12e+00 | 0.02 |
| birkhoff d= 9 | 6.21e-02 | 1.00e+00 | 0.03 |
| cross d= 3 | 1.67e-01 | 8.00e+00 | 0.05 |
| rh_10_25 d= 10 | 1.47e-05 | 3.89e+08 | 0.06 |
| cross d= 8 | 2.48e-05 | 2.56e+02 | 0.06 |
| rh_8_25 d= 8 | 2.35e-04 | 3.34e+06 | 0.09 |
| birkhoff d= 25 | 7.35e-09 | 1.00e+00 | 0.10 |
| cross d= 2 | 5.00e-01 | 4.00e+00 | 0.13 |
| ccp d= 10 | 2.26e-03 | 1.02e+03 | 0.13 |
| cross d= 7 | 1.98e-04 | 1.28e+02 | 0.19 |
| metric d= 10 | 1.18e-04 | 6.05e+07 | 0.27 |
| rh_10_20 d= 10 | 3.84e-06 | 3.61e+09 | 0.28 |
| rh_8_30 d= 8 | 8.03e-04 | 3.08e+05 | 0.32 |
| cc_8_5 d= 8 | 3.01e-06 | 5.76e+02 | 0.46 |
| rh_10_30 d= 10 | 2.71e-05 | 7.45e+07 | 0.51 |
| rh_8_20 d= 8 | 2.24e-07 | 1.68e+11 | 0.53 |
| birkhoff d= 16 | 1.41e-04 | 1.00e+00 | 0.71 |
| cross d= 5 | 8.33e-03 | 3.20e+01 | 0.91 |
| metric d= 6 | 1.88e-03 | 4.67e+04 | 1.54 |
| cc_8_6 d= 8 | 3.16e-06 | 1.41e+05 | 1.95 |
| birkhoff d= 36 | 5.64e-15 | 1.00e+00 | 3.70 |
| birkhoff d= 49 | 4.42e-23 | 1.00e+00 | 13.15 |
| cc_8_8 d= 8 | 2.09e-06 | 3.76e+08 | 295.12 |
| cc_8_9 d= 8 | 6.10e-06 | 2.19e+09 | 1,379.00 |
| cc_8_10 d= 8 | 9.25e-07 | 1.70e+11 | 1,459.43 |
| cc_8_11 d= 8 | 1.69e-10 | 8.25e+15 | 9,085.94 |

Tabla 5.9: Proporción del volumen de P dentro del espacio de referencia.

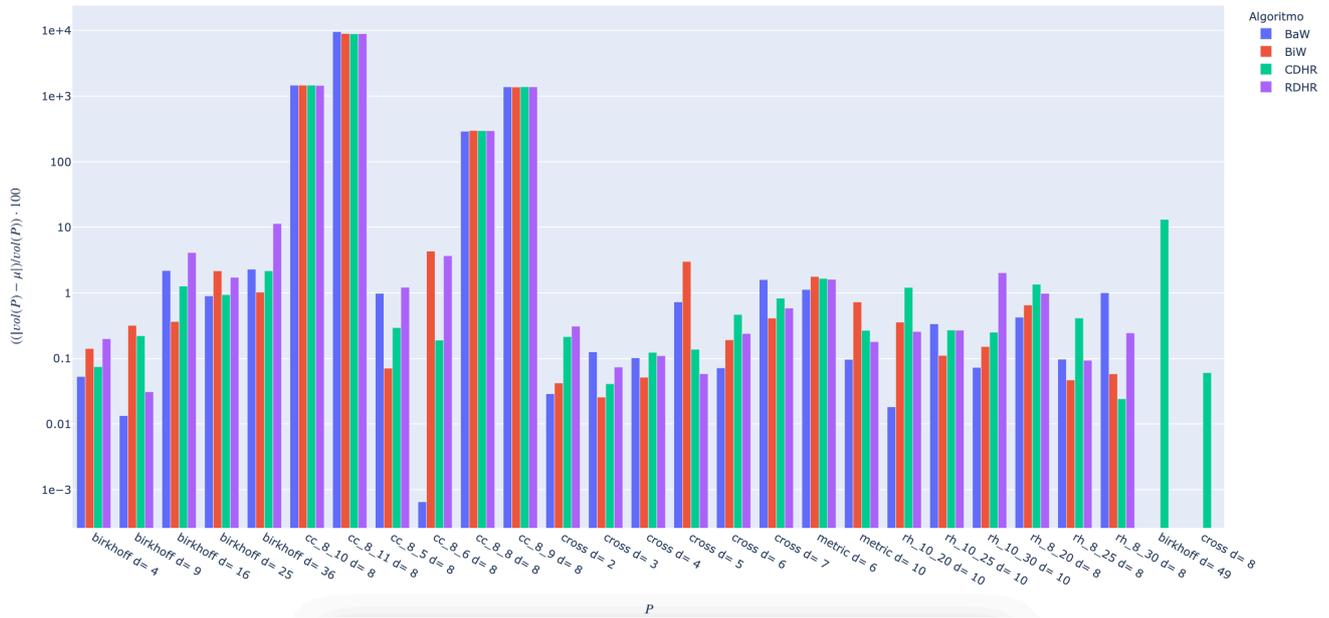
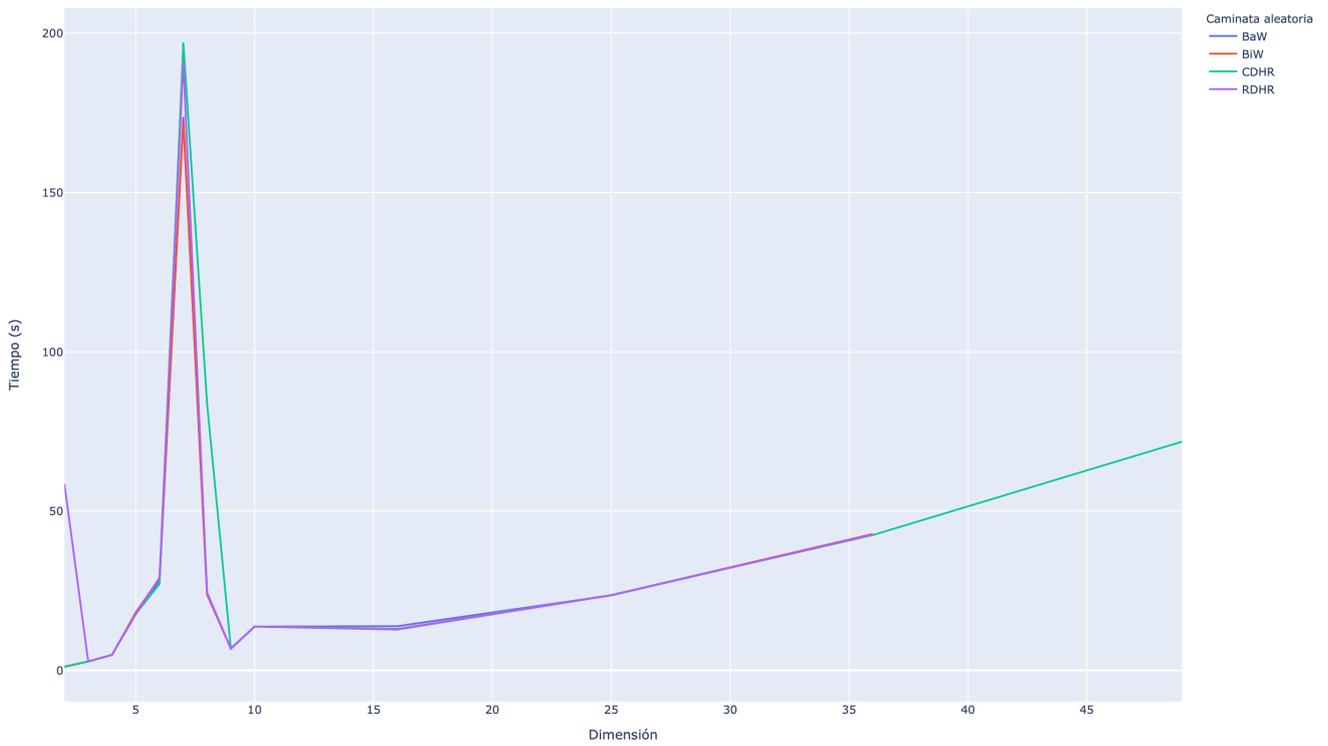
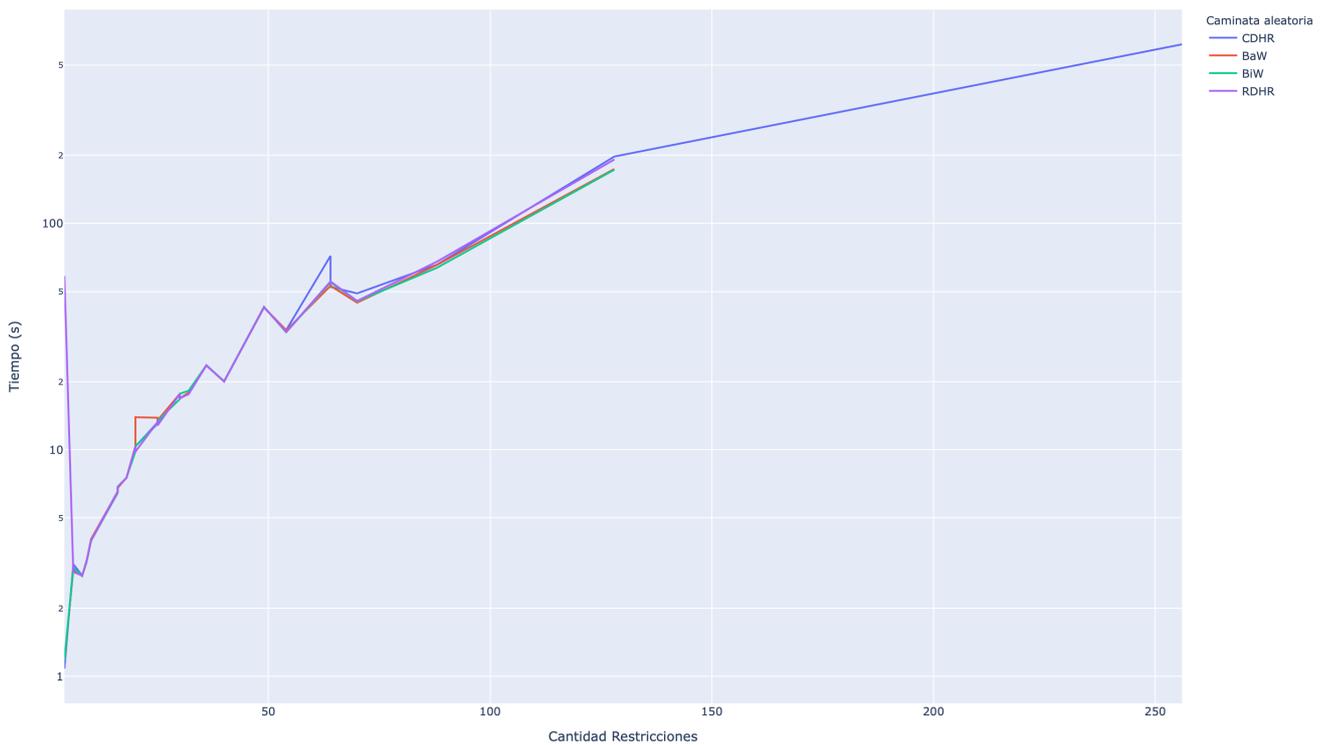


Figura 5.3: Error relativo porcentual calculado para cada politopo. Los datos se presentan en escala logarítmica.

El tiempo de ejecución del algoritmo depende de la dimensión del politopo, como puede verse en la figura 5.4a a medida que aumenta la dimensión aumenta el tiempo de ejecución. Sin embargo, se ve también que la cantidad de restricciones que tiene P , afecta drásticamente el tiempo de ejecución (el pico de la gráfica corresponde al politopo cross 7 que tiene 128 restricciones). Esto es razonable porque por cada restricción que tiene P se agrega una iteración del algoritmo, en la cual se sortean n muestras dentro del espacio de referencia y se verifican cuantos puntos cumplen con todas las restricciones incluidas hasta el momento. En la figura 5.4b se presenta la gráfica de tiempo en función de la cantidad de restricciones, en dicha gráfica se puede ver más fácilmente este comportamiento.



(a) Tiempo de ejecución en función de la dimensión



(b) Tiempo de ejecución en función de la cantidad de restricciones

Figura 5.4: Tiempo de ejecución

Finalmente, teniendo en cuenta los promedios de error y tiempo de ejecución de todos los resultados obtenidos, vemos que la caminata aleatoria con mayor tiempo de ejecución fue RDHR, entre CDHR, BaW y BiW no hubo grandes diferencias, siendo BiW la que mostró un mejor desempeño. Con respecto al error, las cuatro caminatas tuvieron magnitudes similares en el promedio del error, con el siguiente orden descendente: BaW,

BiW, RDHR y CDHR. Se podría decir que a nivel práctico la caminata que mejor rindió fue CDHR, ya que obtuvo el menor promedio de error, en un tiempo de ejecución razonable con respecto a las otras tres. Debido a esto, es que se decidió que la variante de este algoritmo que vamos a utilizar para realizar la experimentación numérica con los modelos metabólicos es la que utiliza la caminata aleatoria *CDHR*.

5.3. Resultados FC

A continuación, en las tablas 5.10 y 5.11, se muestran los resultados obtenidos al aplicar el algoritmo FC en sus dos variantes. Las filas que se presentan en color verde son las filas de las estimaciones que cumplieron el criterio de aceptación que definimos (Error relativo porcentual menor al 10 % y un coeficiente de variación menor a 25 %).

Resultados FC Centro Fijo

| <i>P</i> | <i>m</i> | <i>d</i> | <i>N</i> | Tiempo (s) | Volumen | μ | σ | σ/μ % | <i>min</i> | <i>max</i> | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|------------|----------|----------|----------|----------------|------------|------------|-------------------------|---------|
| birkhoff | 9 | 4 | 1.00e+05 | 4.95e+00 | 1.12e+00 | 1.13e+00 | 6.06e-03 | 0.54 | 1.11e+00 | 1.14e+00 | SI | 0.04 |
| birkhoff | 16 | 9 | 1.00e+05 | 5.18e+00 | 6.21e-02 | 6.07e-02 | 5.27e-03 | 8.69 | 4.82e-02 | 7.50e-02 | SI | 2.27 |
| birkhoff | 25 | 16 | 1.00e+05 | 5.28e+00 | 1.41e-04 | 8.26e-05 | 2.82e-04 | 341.94 | 1.34e-06 | 1.34e-03 | SI | 41.41 |
| birkhoff | 36 | 25 | 1.00e+05 | 5.68e+00 | 7.35e-09 | 2.91e-09 | 3.49e-09 | 119.82 | 3.14e-15 | 1.42e-08 | SI | 60.43 |
| birkhoff | 49 | 36 | 1.00e+05 | 5.99e+00 | 5.64e-15 | 5.60e-15 | 2.92e-14 | 520.57 | 1.02e-15 | 9.80e-16 | NO | 0.63 |
| cc 8 10 | 70 | 8 | 1.00e+05 | 6.20e+00 | 1.57e+05 | 1.30e+06 | 9.21e+05 | 70.61 | 1.04e+06 | 8.64e+05 | NO | 731.76 |
| cc 8 11 | 88 | 8 | 1.00e+05 | 6.71e+00 | 1.39e+06 | 1.45e+07 | 6.54e+06 | 45.11 | 1.03e+07 | 9.86e+06 | NO | 942.12 |
| cc 8 5 | 10 | 8 | 1.00e+05 | 4.99e+00 | 1.74e-03 | 1.59e-03 | 7.04e-04 | 44.20 | 6.53e-04 | 4.26e-03 | SI | 8.29 |
| cc 8 6 | 18 | 8 | 1.00e+05 | 5.21e+00 | 4.44e-01 | 3.55e-01 | 3.08e-01 | 86.90 | 1.65e-01 | 2.17e+00 | SI | 20.21 |
| cc 8 8 | 40 | 8 | 1.00e+05 | 6.05e+00 | 7.84e+02 | 3.39e+03 | 4.05e+03 | 119.37 | 1.15e+03 | 9.80e+02 | NO | 332.58 |
| cc 8 9 | 54 | 8 | 1.00e+05 | 6.48e+00 | 1.33e+04 | 2.25e+05 | 2.06e+05 | 91.81 | 1.23e+05 | 4.12e+05 | SI | 1585.51 |
| ccp | 56 | 10 | 1.00e+05 | 5.97e+00 | 2.31e+00 | 2.31e+00 | 2.57e-02 | 1.11 | 2.27e+00 | 2.40e+00 | SI | 0.01 |
| cross | 4 | 2 | 1.00e+05 | 4.99e+00 | 2.00e+00 | 2.00e+00 | 1.53e-03 | 0.08 | 2.00e+00 | 2.00e+00 | SI | 0.00 |
| cross | 8 | 3 | 1.00e+05 | 5.05e+00 | 1.33e+00 | 1.33e+00 | 1.54e-03 | 0.12 | 1.33e+00 | 1.34e+00 | SI | 0.00 |
| cross | 16 | 4 | 1.00e+05 | 5.19e+00 | 6.67e-01 | 6.66e-01 | 1.04e-03 | 0.16 | 6.64e-01 | 6.69e-01 | SI | 0.03 |
| cross | 32 | 5 | 1.00e+05 | 5.53e+00 | 2.67e-01 | 2.67e-01 | 5.26e-04 | 0.20 | 2.66e-01 | 2.68e-01 | SI | 0.05 |
| cross | 64 | 6 | 1.00e+05 | 6.37e+00 | 8.89e-02 | 8.89e-02 | 2.28e-04 | 0.26 | 8.84e-02 | 8.94e-02 | SI | 0.03 |
| cross | 128 | 7 | 1.00e+05 | 7.80e+00 | 2.54e-02 | 2.54e-02 | 8.20e-05 | 0.32 | 2.52e-02 | 2.56e-02 | SI | 0.01 |
| cross | 256 | 8 | 1.00e+05 | 1.47e+01 | 6.35e-03 | 6.35e-03 | 2.38e-05 | 0.38 | 6.28e-03 | 6.40e-03 | SI | 0.00 |
| cross | 512 | 9 | 1.00e+05 | 1.62e+01 | 1.41e-03 | 1.41e-03 | 6.55e-06 | 0.46 | 1.40e-03 | 1.42e-03 | SI | 0.01 |
| cross | 1,024 | 10 | 1.00e+05 | 2.93e+01 | 2.82e-04 | 2.82e-04 | 1.58e-06 | 0.56 | 2.79e-04 | 2.85e-04 | SI | 0.02 |
| cross | 2,048 | 11 | 1.00e+05 | 5.56e+01 | 5.13e-05 | 5.13e-05 | 2.72e-07 | 0.53 | 5.07e-05 | 5.19e-05 | SI | 0.11 |
| cross | 4,096 | 12 | 1.00e+05 | 1.06e+02 | 8.55e-06 | 8.55e-06 | 5.94e-08 | 0.69 | 8.40e-06 | 8.66e-06 | SI | 0.03 |
| cube | 4 | 2 | 1.00e+05 | 4.64e+00 | 4.00e+00 | 4.00e+00 | 3.01e-03 | 0.08 | 3.99e+00 | 4.01e+00 | SI | 0.01 |
| cube | 6 | 3 | 1.00e+05 | 4.77e+00 | 8.00e+00 | 8.00e+00 | 1.09e-02 | 0.14 | 7.98e+00 | 8.02e+00 | SI | 0.01 |
| cube | 8 | 4 | 1.00e+05 | 4.86e+00 | 1.60e+01 | 1.60e+01 | 2.52e-02 | 0.16 | 1.60e+01 | 1.61e+01 | SI | 0.01 |
| cube | 10 | 5 | 1.00e+05 | 5.00e+00 | 3.20e+01 | 3.20e+01 | 6.76e-02 | 0.21 | 3.19e+01 | 3.21e+01 | SI | 0.10 |
| cube | 12 | 6 | 1.00e+05 | 5.02e+00 | 6.40e+01 | 6.40e+01 | 1.83e-01 | 0.29 | 6.36e+01 | 6.43e+01 | SI | 0.06 |
| cube | 14 | 7 | 1.00e+05 | 5.01e+00 | 1.28e+02 | 1.28e+02 | 4.31e-01 | 0.34 | 1.27e+02 | 1.28e+02 | SI | 0.06 |
| cube | 16 | 8 | 1.00e+05 | 5.10e+00 | 2.64e+02 | 2.56e+02 | 8.98e-01 | 0.35 | 2.54e+02 | 2.57e+02 | SI | 3.06 |
| cube | 18 | 9 | 1.00e+05 | 5.15e+00 | 5.12e+02 | 5.11e+02 | 2.30e+00 | 0.45 | 5.08e+02 | 5.17e+02 | SI | 0.28 |
| cube | 20 | 10 | 1.00e+05 | 5.16e+00 | 1.02e+03 | 1.02e+03 | 5.21e+00 | 0.51 | 1.01e+03 | 1.03e+03 | SI | 0.06 |
| cube | 22 | 11 | 1.00e+05 | 5.07e+00 | 2.05e+03 | 2.05e+03 | 1.15e+01 | 0.56 | 2.02e+03 | 2.06e+03 | SI | 0.09 |
| cube | 24 | 12 | 1.00e+05 | 5.57e+00 | 4.10e+03 | 4.08e+03 | 2.18e+01 | 0.53 | 4.03e+03 | 4.13e+03 | SI | 0.28 |
| cube | 26 | 13 | 1.00e+05 | 5.69e+00 | 8.19e+03 | 8.19e+03 | 5.76e+01 | 0.70 | 8.08e+03 | 8.27e+03 | SI | 0.01 |
| cube | 28 | 14 | 1.00e+05 | 5.72e+00 | 1.64e+04 | 1.63e+04 | 1.07e+02 | 0.66 | 1.61e+04 | 1.65e+04 | SI | 0.34 |
| metric | 6 | 6 | 1.00e+05 | 4.92e+00 | 8.77e+01 | 8.59e+01 | 2.13e+00 | 2.48 | 8.08e+01 | 8.91e+01 | SI | 2.06 |
| metric | 25 | 10 | 1.00e+05 | 5.24e+00 | 7.11e+03 | 7.21e+03 | 1.07e+03 | 14.91 | 1.14e+04 | 8.39e+03 | NO | 1.43 |
| rh 10 20 | 20 | 10 | 1.00e+05 | 5.20e+00 | 1.39e+04 | 1.39e+04 | 7.36e+02 | 5.29 | 1.27e+04 | 1.55e+04 | SI | 0.29 |
| rh 10 25 | 25 | 10 | 1.00e+05 | 5.28e+00 | 5.73e+03 | 5.74e+03 | 1.74e+02 | 3.04 | 5.49e+03 | 6.10e+03 | SI | 0.11 |
| rh 10 30 | 30 | 10 | 1.00e+05 | 5.42e+00 | 2.02e+03 | 2.02e+03 | 2.79e+01 | 1.38 | 1.98e+03 | 2.11e+03 | SI | 0.15 |
| rh 8 20 | 20 | 8 | 1.00e+05 | 5.27e+00 | 3.76e+04 | 2.47e+04 | 1.67e+04 | 67.78 | 1.01e+04 | 7.55e+04 | SI | 34.31 |
| rh 8 25 | 25 | 8 | 1.00e+05 | 5.28e+00 | 7.86e+02 | 7.86e+02 | 1.19e+01 | 1.51 | 7.60e+02 | 8.06e+02 | SI | 0.03 |
| rh 8 30 | 30 | 8 | 1.00e+05 | 5.47e+00 | 2.47e+02 | 2.48e+02 | 1.27e+00 | 0.51 | 2.45e+02 | 2.50e+02 | SI | 0.11 |

Tabla 5.10: Resultados FC con centro fijo

Resultados FC Centro Variable

| P | m | d | N | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $\min \leq \mu \leq \max$ | % Error |
|----------|-------|-----|----------|------------|----------|----------|----------|----------------|----------|----------|---------------------------|---------|
| birkhoff | 9 | 4 | 1.00e+05 | 6.74e+00 | 1.12e+00 | 1.13e+00 | 9.02e-03 | 0.80 | 1.11e+00 | 1.15e+00 | SI | 0.09 |
| birkhoff | 16 | 9 | 1.00e+05 | 7.07e+00 | 6.21e-02 | 6.22e-02 | 6.19e-03 | 9.96 | 4.92e-02 | 7.70e-02 | SI | 0.11 |
| birkhoff | 25 | 16 | 1.00e+05 | 7.07e+00 | 1.41e-04 | 9.98e-05 | 6.99e-05 | 70.06 | 2.22e-05 | 3.07e-04 | SI | 29.18 |
| birkhoff | 36 | 25 | 1.00e+05 | 7.48e+00 | 7.35e-09 | 3.84e-08 | 2.34e-07 | 609.82 | 6.75e-12 | 1.50e-06 | SI | 421.72 |
| birkhoff | 49 | 36 | 1.00e+05 | 8.04e+00 | 5.64e-15 | 1.53e-17 | 7.04e-17 | 459.01 | 1.02e-19 | 9.35e-21 | NO | 99.73 |
| cc 8 10 | 70 | 8 | 1.00e+05 | 7.91e+00 | 1.57e+05 | 1.92e+06 | 2.68e+06 | 139.42 | 1.01e+06 | 9.69e+05 | NO | 1123.72 |
| cc 8 11 | 88 | 8 | 1.00e+05 | 6.61e+00 | 1.39e+06 | 2.35e+07 | 7.03e+06 | 29.90 | 1.99e+07 | 9.77e+06 | NO | 1589.41 |
| cc 8 5 | 10 | 8 | 1.00e+05 | 6.79e+00 | 1.74e-03 | 1.74e-03 | 1.59e-03 | 91.31 | 3.80e-04 | 9.22e-03 | SI | 0.41 |
| cc 8 6 | 18 | 8 | 1.00e+05 | 7.31e+00 | 4.44e-01 | 4.25e-01 | 3.33e-01 | 78.39 | 7.77e-02 | 1.65e+00 | SI | 4.48 |
| cc 8 8 | 40 | 8 | 1.00e+05 | 7.66e+00 | 7.84e+02 | 4.36e+03 | 9.73e+03 | 223.22 | 1.06e+03 | 9.96e+02 | NO | 456.04 |
| cc 8 9 | 54 | 8 | 1.00e+05 | 7.60e+00 | 1.33e+04 | 1.83e+05 | 4.35e+04 | 23.79 | 1.14e+05 | 3.52e+05 | SI | 1270.60 |
| ccp | 56 | 10 | 1.00e+05 | 7.80e+00 | 2.31e+00 | 2.33e+00 | 1.03e-01 | 4.40 | 2.08e+00 | 2.51e+00 | SI | 0.86 |
| cross | 4 | 2 | 1.00e+05 | 7.13e+00 | 2.00e+00 | 2.00e+00 | 3.01e-03 | 0.15 | 1.99e+00 | 2.01e+00 | SI | 0.01 |
| cross | 8 | 3 | 1.00e+05 | 7.68e+00 | 1.33e+00 | 1.33e+00 | 3.07e-03 | 0.23 | 1.33e+00 | 1.34e+00 | SI | 0.01 |
| cross | 16 | 4 | 1.00e+05 | 7.68e+00 | 6.67e-01 | 6.67e-01 | 2.40e-03 | 0.36 | 6.62e-01 | 6.72e-01 | SI | 0.03 |
| cross | 32 | 5 | 1.00e+05 | 7.77e+00 | 2.67e-01 | 2.66e-01 | 1.60e-03 | 0.60 | 2.64e-01 | 2.71e-01 | SI | 0.09 |
| cross | 64 | 6 | 1.00e+05 | 8.26e+00 | 8.89e-02 | 8.88e-02 | 6.36e-04 | 0.72 | 8.74e-02 | 9.02e-02 | SI | 0.12 |
| cross | 128 | 7 | 1.00e+05 | 9.28e+00 | 2.54e-02 | 2.54e-02 | 2.80e-04 | 1.10 | 2.48e-02 | 2.62e-02 | SI | 0.08 |
| cross | 256 | 8 | 1.00e+05 | 1.21e+01 | 6.35e-03 | 6.35e-03 | 8.70e-05 | 1.37 | 6.14e-03 | 6.55e-03 | SI | 0.05 |
| cross | 512 | 9 | 1.00e+05 | 1.82e+01 | 1.41e-03 | 1.41e-03 | 2.58e-05 | 1.84 | 1.36e-03 | 1.46e-03 | SI | 0.42 |
| cross | 1,024 | 10 | 1.00e+05 | 3.17e+01 | 2.82e-04 | 2.82e-04 | 6.55e-06 | 2.32 | 2.69e-04 | 2.93e-04 | SI | 0.11 |
| cross | 2,048 | 11 | 1.00e+05 | 5.64e+01 | 5.13e-05 | 5.17e-05 | 2.34e-06 | 4.52 | 4.75e-05 | 5.61e-05 | SI | 0.67 |
| cross | 4,096 | 12 | 1.00e+05 | 9.82e+01 | 8.55e-06 | 8.67e-06 | 4.82e-07 | 5.55 | 7.90e-06 | 9.58e-06 | SI | 1.39 |
| cube | 4 | 2 | 1.00e+05 | 6.72e+00 | 4.00e+00 | 4.00e+00 | 4.38e-03 | 0.11 | 3.99e+00 | 4.01e+00 | SI | 0.01 |
| cube | 6 | 3 | 1.00e+05 | 6.65e+00 | 8.00e+00 | 8.00e+00 | 2.18e-02 | 0.27 | 7.95e+00 | 8.04e+00 | SI | 0.00 |
| cube | 8 | 4 | 1.00e+05 | 6.50e+00 | 1.60e+01 | 1.60e+01 | 5.77e-02 | 0.36 | 1.59e+01 | 1.61e+01 | SI | 0.04 |
| cube | 10 | 5 | 1.00e+05 | 6.53e+00 | 3.20e+01 | 3.20e+01 | 1.81e-01 | 0.56 | 3.15e+01 | 3.23e+01 | SI | 0.06 |
| cube | 12 | 6 | 1.00e+05 | 6.73e+00 | 6.40e+01 | 6.39e+01 | 5.75e-01 | 0.90 | 6.27e+01 | 6.55e+01 | SI | 0.12 |
| cube | 14 | 7 | 1.00e+05 | 6.72e+00 | 1.28e+02 | 1.28e+02 | 1.73e+00 | 1.35 | 1.24e+02 | 1.31e+02 | SI | 0.10 |
| cube | 16 | 8 | 1.00e+05 | 6.62e+00 | 2.64e+02 | 2.56e+02 | 4.63e+00 | 1.81 | 2.49e+02 | 2.64e+02 | SI | 3.15 |
| cube | 18 | 9 | 1.00e+05 | 6.76e+00 | 5.12e+02 | 5.07e+02 | 1.12e+01 | 2.21 | 4.85e+02 | 5.32e+02 | SI | 0.91 |
| cube | 20 | 10 | 1.00e+05 | 6.90e+00 | 1.02e+03 | 1.02e+03 | 2.36e+01 | 2.31 | 1.00e+03 | 9.96e+02 | NO | 0.18 |
| cube | 22 | 11 | 1.00e+05 | 6.85e+00 | 2.05e+03 | 2.03e+03 | 1.04e+02 | 5.10 | 1.84e+03 | 2.26e+03 | SI | 0.69 |
| cube | 24 | 12 | 1.00e+05 | 7.00e+00 | 4.10e+03 | 4.05e+03 | 2.08e+02 | 5.14 | 3.69e+03 | 4.51e+03 | SI | 1.18 |
| cube | 26 | 13 | 1.00e+05 | 7.11e+00 | 8.19e+03 | 8.02e+03 | 5.47e+02 | 6.82 | 7.06e+03 | 9.18e+03 | SI | 2.05 |
| cube | 28 | 14 | 1.00e+05 | 6.93e+00 | 1.64e+04 | 1.67e+04 | 1.96e+03 | 11.77 | 1.40e+04 | 2.08e+04 | SI | 1.70 |
| metric | 6 | 6 | 1.00e+05 | 6.58e+00 | 8.77e+01 | 8.60e+01 | 3.82e+00 | 4.44 | 8.03e+01 | 9.61e+01 | SI | 1.92 |
| metric | 25 | 10 | 1.00e+05 | 7.34e+00 | 7.11e+03 | 7.21e+03 | 9.90e+02 | 13.72 | 5.37e+03 | 9.63e+03 | SI | 1.43 |
| rh 10 20 | 20 | 10 | 1.00e+05 | 6.99e+00 | 1.39e+04 | 1.32e+04 | 2.23e+03 | 16.87 | 1.01e+04 | 1.74e+04 | SI | 4.63 |
| rh 10 25 | 25 | 10 | 1.00e+05 | 7.06e+00 | 5.73e+03 | 5.67e+03 | 9.17e+02 | 16.17 | 4.33e+03 | 8.92e+03 | SI | 1.06 |
| rh 10 30 | 30 | 10 | 1.00e+05 | 7.21e+00 | 2.02e+03 | 2.00e+03 | 1.38e+02 | 6.89 | 1.77e+03 | 2.30e+03 | SI | 0.84 |
| rh 8 20 | 20 | 8 | 1.00e+05 | 6.86e+00 | 3.76e+04 | 3.10e+04 | 1.96e+04 | 63.27 | 1.55e+04 | 9.18e+04 | SI | 17.57 |
| rh 8 25 | 25 | 8 | 1.00e+05 | 6.91e+00 | 7.86e+02 | 7.78e+02 | 5.22e+01 | 6.70 | 7.16e+02 | 9.40e+02 | SI | 1.00 |
| rh 8 30 | 30 | 8 | 1.00e+05 | 7.46e+00 | 2.47e+02 | 2.48e+02 | 7.14e+00 | 2.88 | 2.32e+02 | 2.61e+02 | SI | 0.35 |

Tabla 5.11: Resultados FC con centro variable

5.3.1. Análisis de resultados

En las figuras 5.5 y 5.7 se puede ver la evolución del error de estimación y el tiempo de ejecución en función de la dimensión del polítopo respectivamente. Los datos se presentan discriminados, según si la distancia del punto interior a los bordes del polítopo, se calcula a partir de un punto fijo o de puntos variables, calculados en cada iteración del algoritmo.

Cuando el punto interior es variable, se puede ver en la figura 5.5, que se pierde calidad del estimador. Esto puede estar ocurriendo debido a la variación en la distribución de las distancias entre un punto interior y el borde del polítopo.

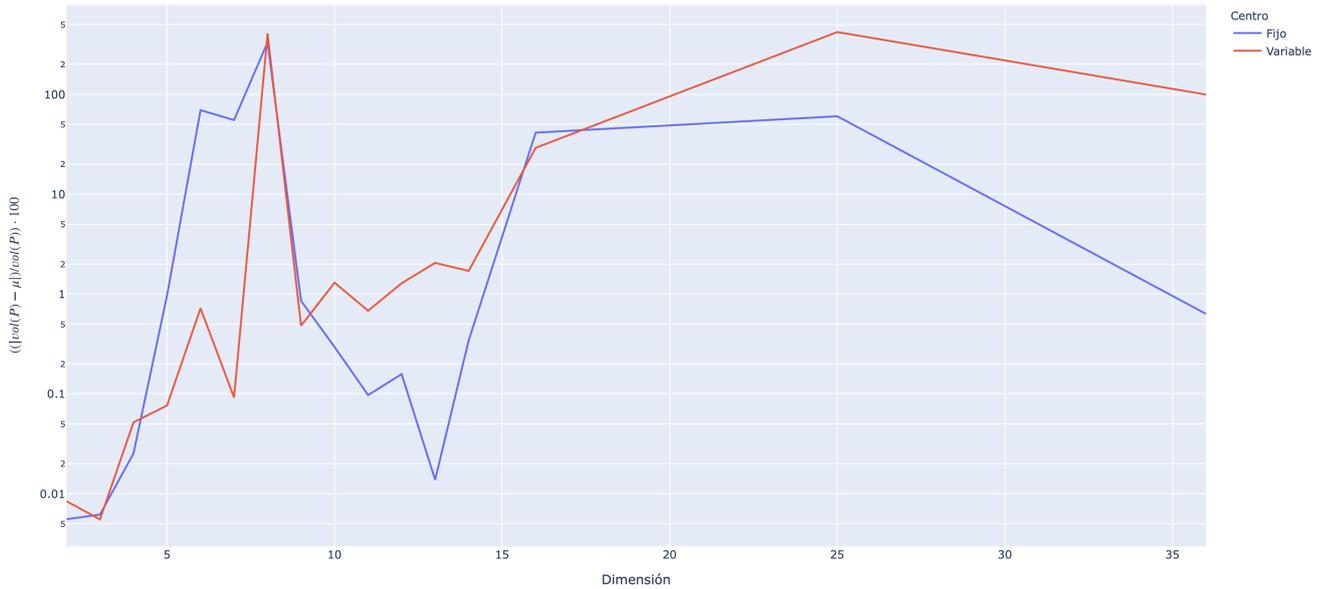


Figura 5.5: Error en función de la dimensión. Los datos se presentan en escala logarítmica.

En la figura 5.6, se muestra el promedio de error relativo para cada uno de los polítopos. Aquí se puede ver que en la mayoría de los polítopos, el error relativo porcentual fue menor cuando utilizamos la variante del algoritmo con centro fijo.

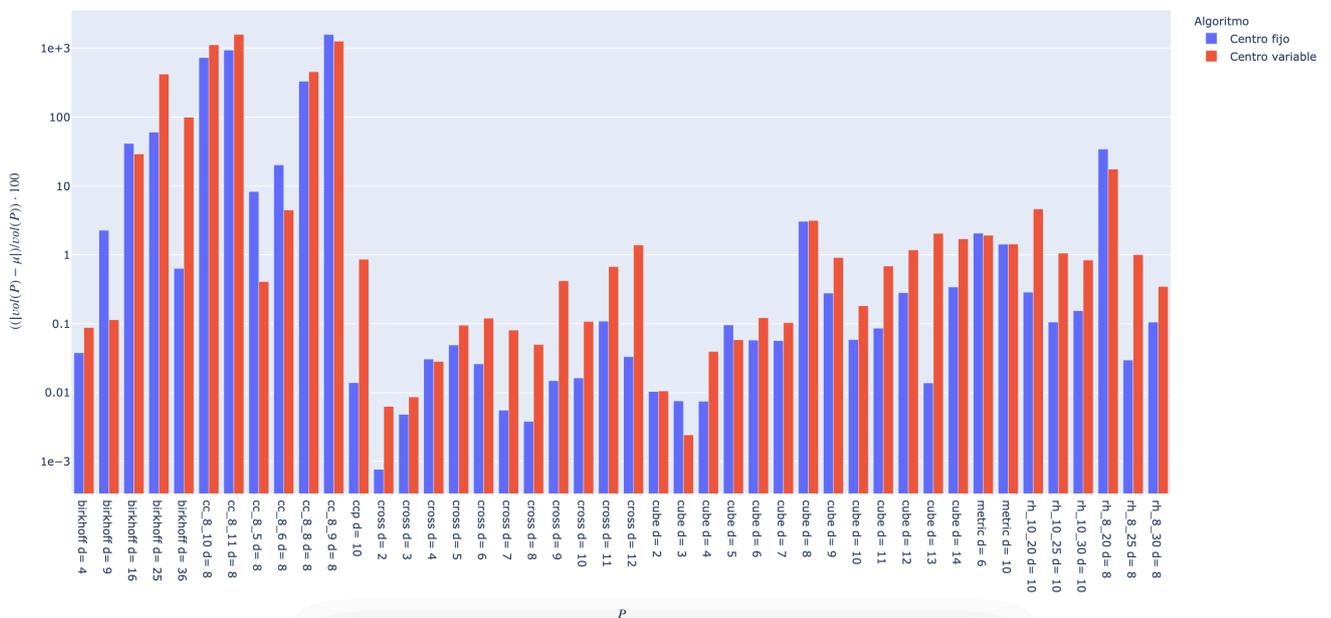
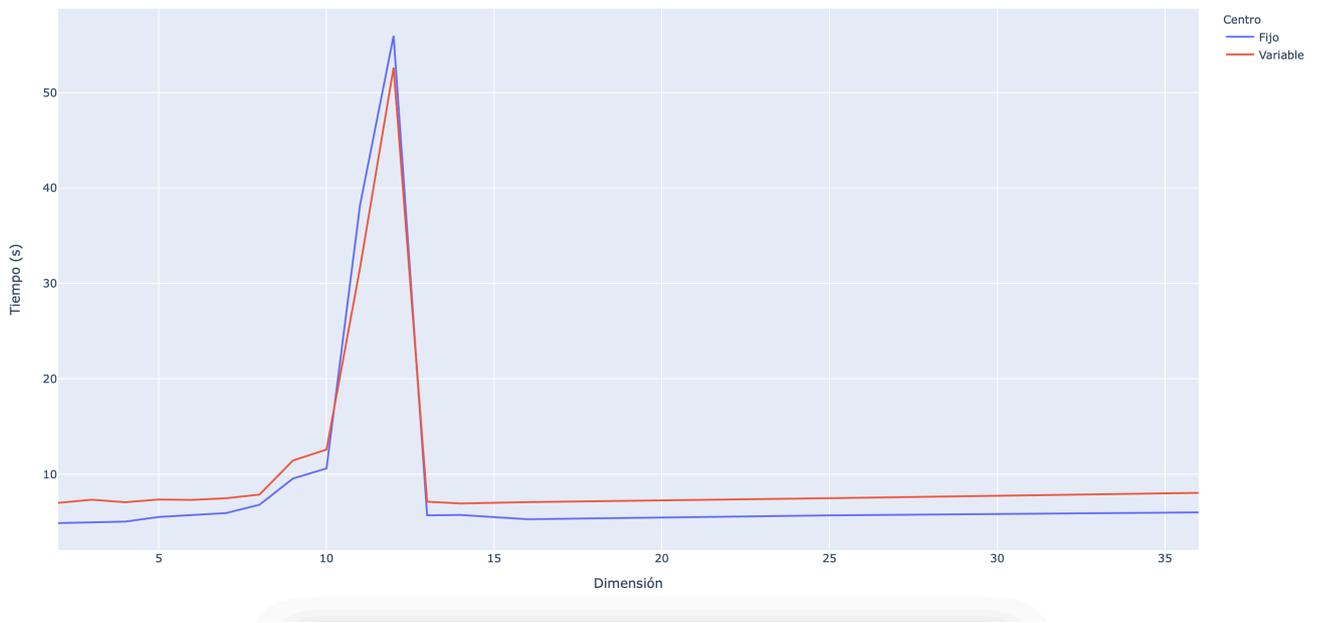
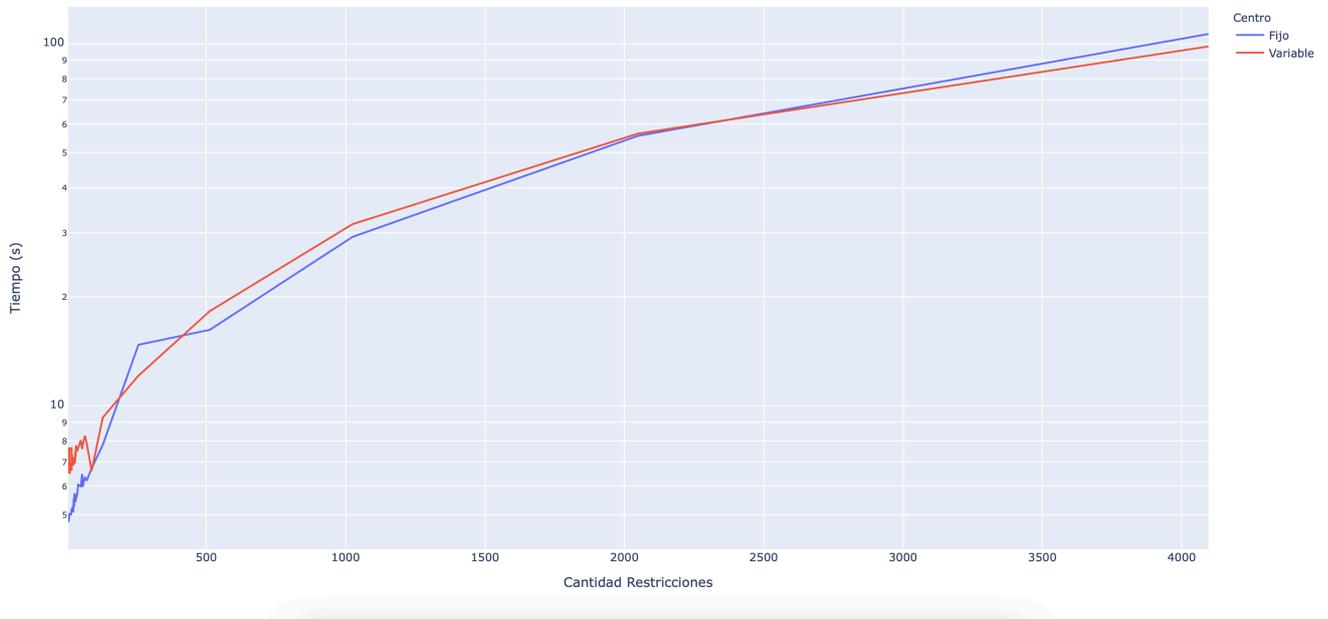


Figura 5.6: Promedio de error relativo por cada polítopo. Los datos se presentan en escala logarítmica.

Con respecto al tiempo de ejecución, en la figura 5.7 se ve que en estos casos, depende más de la cantidad de restricciones que definen a P , que de la dimensión del mismo. Esto es porque la cantidad de restricciones impacta en el tiempo que lleva hacer el cálculo para encontrar los bordes de P a partir de un punto interior.



(a) Tiempo de ejecución en función de la dimensión.



(b) Tiempo de ejecución en función de la cantidad de restricciones de P .

Figura 5.7: Tiempo de ejecución y error FC. Los datos se presentan en escala logarítmica.

En general el tiempo de ejecución y el promedio de error para ambas variantes del algoritmo resultaron bastante similares. Pero como pudimos ver en la figura 5.6, en la mayoría de los polítopos obtuvimos un error relativo porcentual menor utilizando la variante del algoritmo de centro fijo, por esto es que se decidió utilizar dicha variante para el estudio de los modelos metabólicos.

5.4. Resultados MCMC

A continuación, en las tablas 5.12 y 5.13, se muestran los resultados obtenidos al aplicar el algoritmo basado en MCMC. La cantidad de cadenas b y de muestreos N que son necesarios como parámetros del algoritmo fueron determinados experimentalmente. Las filas que se presentan en color verde son las filas de las estimaciones que cumplieron el criterio de aceptación que definimos (Error relativo porcentual menor al 10% y un coeficiente de variación menor a 25%).

| <i>P</i> | <i>m</i> | <i>d</i> | <i>N</i> | <i>b</i> | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|----------|------------|----------|----------|----------|----------------|----------|----------|-------------------------|---------|
| birkhoff | 9 | 4 | 1000 | 4 | 7.90e+00 | 1.12e+00 | 1.23e+00 | 1.06e-01 | 8.63 | 1.03e+00 | 1.47e+00 | SI | 9.53 |
| birkhoff | 16 | 9 | 1000 | 10 | 2.66e+01 | 6.21e-02 | 2.60e-02 | 6.05e-03 | 23.24 | 1.89e-02 | 4.51e-02 | SI | 58.08 |
| birkhoff | 25 | 16 | 1000 | 18 | 6.93e+01 | 1.41e-04 | 4.47e-06 | 2.28e-06 | 50.89 | 1.75e-06 | 1.29e-05 | SI | 96.82 |
| birkhoff | 36 | 25 | 1000 | 28 | 2.07e+02 | 7.35e-09 | 4.13e-10 | 9.12e-11 | 22.07 | 2.95e-10 | 6.64e-10 | SI | 94.38 |
| birkhoff | 49 | 36 | 1000 | 40 | 4.24e+02 | 5.64e-15 | 1.81e-17 | 6.54e-18 | 36.21 | 1.06e-17 | 9.36e-18 | NO | 99.68 |
| birkhoff | 64 | 49 | 1000 | 54 | 9.27e+02 | 4.42e-23 | 1.96e-27 | 1.06e-27 | 54.19 | 1.08e-27 | 8.67e-28 | NO | 100.00 |
| birkhoff | 81 | 64 | 1000 | 70 | 1.55e+03 | 2.60e-33 | 2.49e-40 | 1.47e-40 | 58.94 | 1.00e-40 | 9.37e-41 | NO | 100.00 |
| birkhoff | 100 | 81 | 1000 | 90 | 3.24e+03 | 8.78e-46 | 1.67e-56 | 8.94e-57 | 53.36 | 1.03e-56 | 9.13e-57 | NO | 100.00 |
| cc_8_10 | 70 | 8 | 1000 | 10 | 3.14e+01 | 1.57e+05 | 8.58e+05 | 3.92e+05 | 45.68 | 1.01e+06 | 9.78e+05 | NO | 447.25 |
| cc_8_11 | 88 | 8 | 1000 | 10 | 3.81e+01 | 1.39e+06 | 1.08e+07 | 1.98e+07 | 182.61 | 1.93e+06 | 9.11e+07 | SI | 678.15 |
| cc_8_5 | 10 | 8 | 1000 | 10 | 2.94e+01 | 1.74e-03 | 7.98e-04 | 4.89e-04 | 61.27 | 3.59e-04 | 4.52e-03 | SI | 54.04 |
| cc_8_6 | 18 | 8 | 1000 | 10 | 2.75e+01 | 4.44e-01 | 2.76e-01 | 4.97e-01 | 179.81 | 1.05e-01 | 4.33e+00 | SI | 37.80 |
| cc_8_8 | 40 | 8 | 1000 | 10 | 2.73e+01 | 7.84e+02 | 2.67e+03 | 1.23e+03 | 46.24 | 1.35e+03 | 5.66e+03 | SI | 239.98 |
| cc_8_9 | 54 | 8 | 1000 | 10 | 2.68e+01 | 1.33e+04 | 8.29e+04 | 3.22e+04 | 38.83 | 1.00e+05 | 9.66e+04 | NO | 521.41 |
| ccp | 56 | 10 | 1000 | 16 | 5.80e+01 | 2.31e+00 | 1.83e+00 | 8.11e-02 | 4.43 | 1.69e+00 | 2.09e+00 | SI | 20.80 |
| ccp | 56 | 10 | 1000 | 12 | 5.02e+01 | 2.31e+00 | 1.84e+00 | 1.09e-01 | 5.94 | 1.72e+00 | 2.12e+00 | SI | 20.49 |
| ccp | 56 | 10 | 1000 | 10 | 3.10e+01 | 2.31e+00 | 1.96e+00 | 1.63e-01 | 8.34 | 1.83e+00 | 2.24e+00 | SI | 15.36 |
| ccp | 56 | 10 | 1000 | 30 | 1.90e+02 | 2.31e+00 | 1.87e+00 | 0.00e+00 | 0.00 | 1.87e+00 | 1.87e+00 | SI | 19.19 |
| cross | 4 | 2 | 1000 | 6 | 1.42e+01 | 2.00e+00 | 2.00e+00 | 7.55e-03 | 0.38 | 1.99e+00 | 2.03e+00 | SI | 0.16 |
| cross | 4 | 2 | 1000 | 2 | 7.51e+00 | 2.00e+00 | 2.01e+00 | 1.08e-02 | 0.54 | 2.00e+00 | 2.02e+00 | SI | 0.55 |
| cross | 8 | 3 | 1000 | 4 | 9.54e+00 | 1.33e+00 | 1.35e+00 | 1.21e-02 | 0.90 | 1.32e+00 | 1.38e+00 | SI | 1.13 |
| cross | 16 | 4 | 1000 | 4 | 8.86e+00 | 6.67e-01 | 6.81e-01 | 9.10e-03 | 1.34 | 6.63e-01 | 7.09e-01 | SI | 2.08 |
| cross | 32 | 5 | 1000 | 6 | 1.43e+01 | 2.67e-01 | 2.69e-01 | 3.91e-03 | 1.45 | 2.61e-01 | 2.79e-01 | SI | 1.03 |
| cross | 64 | 6 | 1000 | 8 | 2.24e+01 | 8.89e-02 | 8.92e-02 | 1.43e-03 | 1.61 | 8.60e-02 | 9.22e-02 | SI | 0.36 |
| cross | 128 | 7 | 1000 | 8 | 2.14e+01 | 2.54e-02 | 2.56e-02 | 4.74e-04 | 1.85 | 2.48e-02 | 2.73e-02 | SI | 0.85 |
| cross | 256 | 8 | 1000 | 10 | 2.92e+01 | 6.35e-03 | 6.40e-03 | 1.14e-04 | 1.79 | 6.15e-03 | 6.63e-03 | SI | 0.83 |

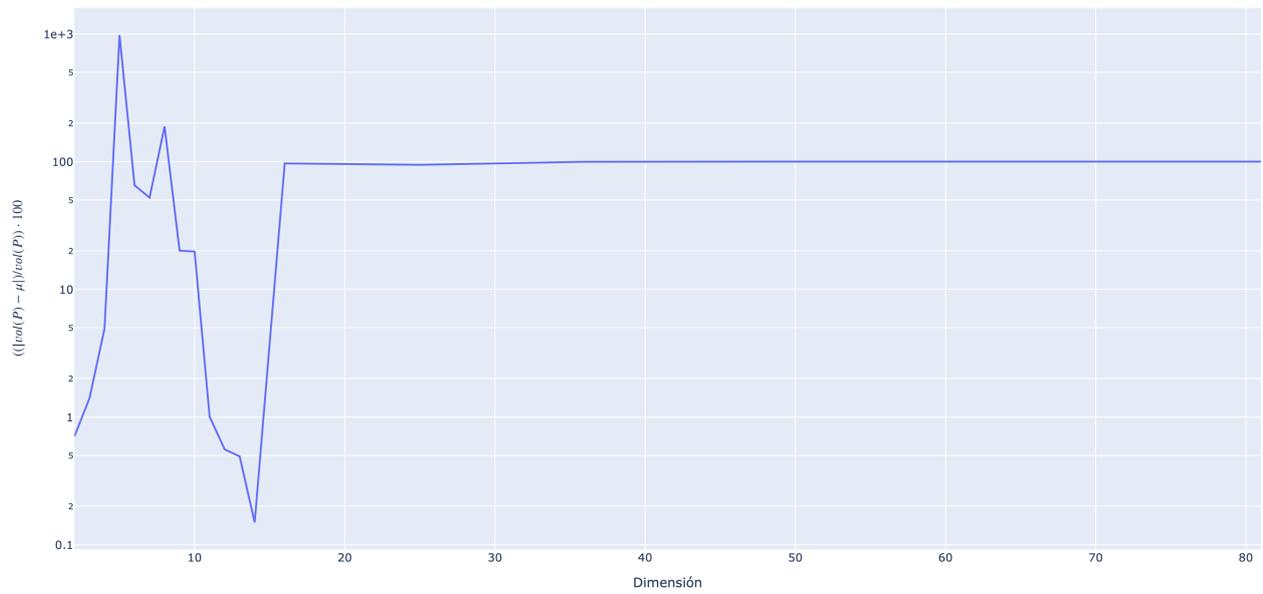
Tabla 5.12: Resultados MCMC

| <i>P</i> | <i>m</i> | <i>d</i> | <i>N</i> | <i>b</i> | Tiempo (s) | Volumen | μ | σ | σ/μ % | min | max | $min \leq \mu \leq max$ | % Error |
|----------|----------|----------|----------|----------|------------|----------|----------|----------|----------------|----------|----------|-------------------------|---------|
| cross | 1,024 | 10 | 1000 | 12 | 3.86e+01 | 2.82e-04 | 2.80e-04 | 6.05e-06 | 2.16 | 2.67e-04 | 2.92e-04 | SI | 0.72 |
| cross | 2,048 | 11 | 1000 | 12 | 4.11e+01 | 5.13e-05 | 5.11e-05 | 1.22e-06 | 2.38 | 4.86e-05 | 5.36e-05 | SI | 0.38 |
| cross | 4,096 | 12 | 1000 | 14 | 8.09e+01 | 8.55e-06 | 8.48e-06 | 2.31e-07 | 2.73 | 8.00e-06 | 9.00e-06 | SI | 0.89 |
| cube | 4 | 2 | 1000 | 2 | 4.56e+00 | 4.00e+00 | 4.05e+00 | 2.99e-02 | 0.74 | 4.00e+00 | 4.11e+00 | SI | 1.24 |
| cube | 6 | 3 | 1000 | 4 | 8.09e+00 | 8.00e+00 | 8.13e+00 | 6.96e-02 | 0.86 | 7.97e+00 | 8.25e+00 | SI | 1.68 |
| cube | 8 | 4 | 1000 | 4 | 7.84e+00 | 1.60e+01 | 1.65e+01 | 2.66e-01 | 1.61 | 1.61e+01 | 1.72e+01 | SI | 3.13 |
| cube | 10 | 5 | 1000 | 6 | 1.29e+01 | 3.20e+01 | 3.26e+01 | 5.86e-01 | 1.80 | 3.16e+01 | 3.40e+01 | SI | 1.93 |
| cube | 12 | 6 | 1000 | 8 | 1.89e+01 | 6.40e+01 | 6.52e+01 | 1.03e+00 | 1.59 | 6.32e+01 | 6.68e+01 | SI | 1.86 |
| cube | 14 | 7 | 1000 | 8 | 1.92e+01 | 1.28e+02 | 1.31e+02 | 2.19e+00 | 1.67 | 1.28e+02 | 1.35e+02 | SI | 2.29 |
| cube | 16 | 8 | 1000 | 10 | 2.71e+01 | 2.64e+02 | 2.60e+02 | 6.54e+00 | 2.52 | 2.48e+02 | 2.73e+02 | SI | 1.61 |
| cube | 18 | 9 | 1000 | 10 | 2.67e+01 | 5.12e+02 | 5.20e+02 | 1.34e+01 | 2.58 | 4.95e+02 | 5.45e+02 | SI | 1.60 |
| cube | 20 | 10 | 1000 | 12 | 3.49e+01 | 1.02e+03 | 1.02e+03 | 2.97e+01 | 2.91 | 1.00e+03 | 9.99e+02 | NO | 0.29 |
| cube | 22 | 11 | 1000 | 12 | 3.51e+01 | 2.05e+03 | 2.08e+03 | 4.82e+01 | 2.32 | 2.00e+03 | 2.19e+03 | SI | 1.63 |
| cube | 24 | 12 | 1000 | 14 | 4.44e+01 | 4.10e+03 | 4.11e+03 | 1.59e+02 | 3.89 | 3.72e+03 | 4.38e+03 | SI | 0.23 |
| cube | 26 | 13 | 1000 | 14 | 4.51e+01 | 8.19e+03 | 8.23e+03 | 2.40e+02 | 2.92 | 7.70e+03 | 8.61e+03 | SI | 0.49 |
| cube | 28 | 14 | 1000 | 16 | 5.58e+01 | 1.64e+04 | 1.64e+04 | 5.31e+02 | 3.24 | 1.52e+04 | 1.73e+04 | SI | 0.15 |
| metric | 6 | 6 | 1000 | 8 | 2.21e+01 | 8.77e+01 | 5.14e+01 | 6.83e+00 | 13.29 | 3.94e+01 | 6.52e+01 | SI | 41.44 |
| metric | 25 | 10 | 1000 | 12 | 3.81e+01 | 7.11e+03 | 3.58e+03 | 2.43e+02 | 6.79 | 2.94e+03 | 4.00e+03 | SI | 49.68 |
| rh_10_20 | 20 | 10 | 1000 | 12 | 3.79e+01 | 1.39e+04 | 1.00e+04 | 1.33e+03 | 13.19 | 1.01e+04 | 9.84e+03 | NO | 27.64 |
| rh_10_25 | 25 | 10 | 1000 | 12 | 3.51e+01 | 5.73e+03 | 4.35e+03 | 3.05e+02 | 7.03 | 3.95e+03 | 5.28e+03 | SI | 24.14 |
| rh_10_30 | 30 | 10 | 1000 | 12 | 3.51e+01 | 2.02e+03 | 1.70e+03 | 1.09e+02 | 6.39 | 1.54e+03 | 1.89e+03 | SI | 15.51 |
| rh_8_20 | 20 | 8 | 1000 | 10 | 3.00e+01 | 3.76e+04 | 1.37e+04 | 8.58e+03 | 62.42 | 1.10e+04 | 9.40e+03 | NO | 63.41 |
| rh_8_25 | 25 | 8 | 1000 | 10 | 2.82e+01 | 7.86e+02 | 6.27e+02 | 5.91e+01 | 9.41 | 5.30e+02 | 7.77e+02 | SI | 20.18 |
| rh_8_30 | 30 | 8 | 1000 | 10 | 2.82e+01 | 2.47e+02 | 2.32e+02 | 9.57e+00 | 4.12 | 2.15e+02 | 2.48e+02 | SI | 6.11 |

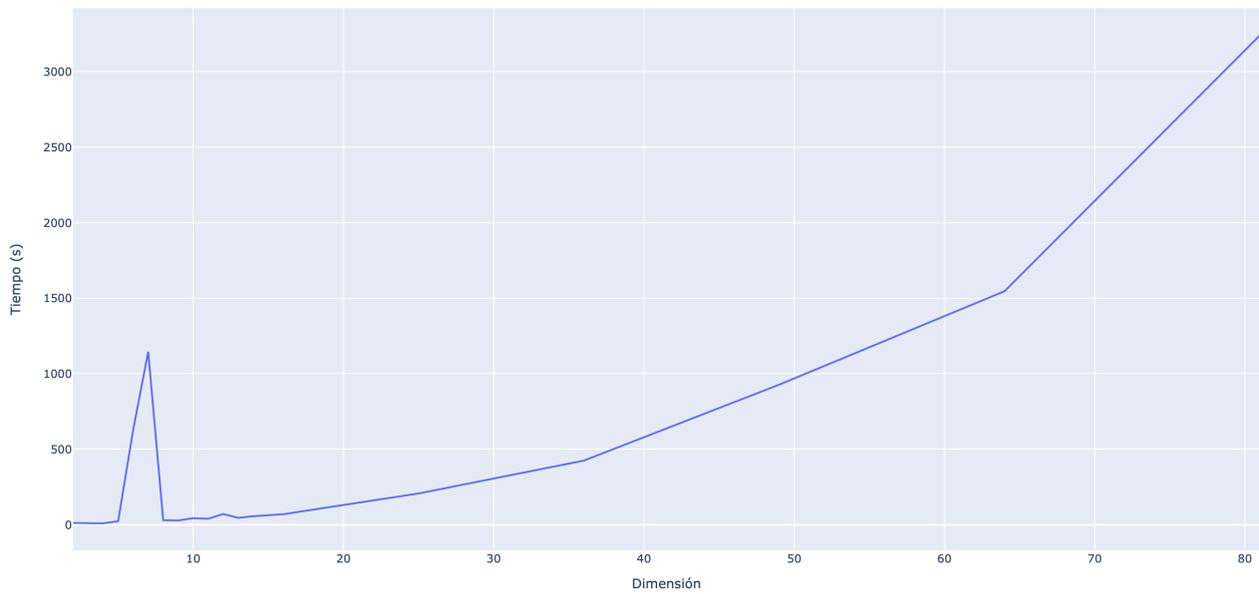
Tabla 5.13: Resultados MCMC

5.4.1. Análisis de resultados

En las gráficas de la figura 5.8 se puede ver la evolución del error de estimación y el tiempo de ejecución en función de la dimensión del politopo respectivamente.



(a) Error en función de la dimensión.



(b) Tiempo de ejecución en función de la dimensión

Figura 5.8: Error y tiempo de ejecución en función de la dimensión. Los datos se presentan en escala logarítmica.

En relación al tiempo de ejecución, se puede ver en la grafica de la figura 5.8b que a medida que aumenta la dimensión de P hay un notorio aumento en el tiempo de ejecución del algoritmo.

Con respecto al error, vemos que en las dimensiones más bajas, los politopos que tuvieron mayor error fueron los llamados cc_n_m , igual que en los metodos anteriores. En la gráfica de la figura 5.9 podemos ver el error relativo por cada politopo. Por otra parte, a diferencia de los algoritmos anteriores, en este se pudo realizar las pruebas con politopos de dimensión más grande. Alcanzando dimensión 81 con el politopo de Birkhoff ([1] son politopos que cumplen que la suma de los valores de sus filas y columnas es igual a uno). Aunque fue posible aplicar el algoritmo en estas dimensiones, vemos que los resultados obtenidos no pueden ser tomados en cuenta, ya que arrojaron un error relativo porcentual de 100%.

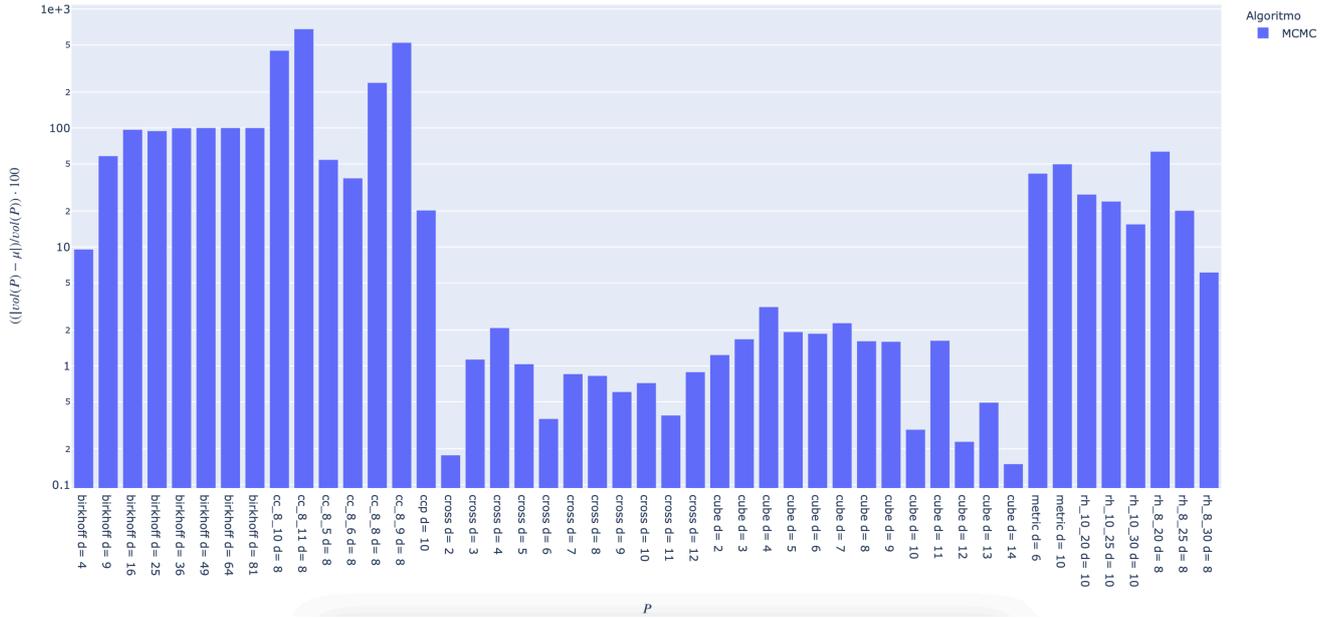


Figura 5.9: Error relativo, por cada politopo. Los datos se presentan en escala logarítmica.

5.4.2. Análisis de los resultados obtenidos

Luego de aplicar los algoritmos estudiados vemos que en todos los casos los politopos que tuvieron mayor error relativo son los cc_m_n . Para entender mejor la forma de los politopos que estamos estudiando, y entender si la forma de los mismos es la que está afectando los resultados, nos basamos en la idea desarrollada en [4]. Esta idea es la que se aplica para redondear los politopos antes de aplicar el algoritmo CB para aproximar el volumen (se puede consultar una descripción más detallada en A.3). En el artículo, se describe cómo se calcula una transformación lineal que “redondea” a P para evitar los problemas causados por formas alargadas. Para calcular esta transformación, se sortean muestras utilizando alguna caminata aleatoria dentro del politopo y se hace la descomposición en valores singulares de estos puntos. Es decir, para la matriz de puntos M , se encuentran matrices tales que $M = USV'$ con $M \in R^{m \times n}$ tal que S es una matriz diagonal que contiene los n valores singulares s_i , $U \in R^{m \times n}$ y $V \in R^{n \times n}$ cuyas columnas forman una base ortonormal de R^m y R^n respectivamente. La idea que explican los autores de este algoritmo, y que nos interesa para intentar entender la forma de los politopos con los que estamos trabajando, es que los valores singulares s_i de esta matriz de puntos, representan la coordenada de P en la dirección $u_i v_i^T$. Por lo que comparando estos valores singulares, podemos ver si P tiene alguna dirección que sea considerablemente más “larga” que las demás.

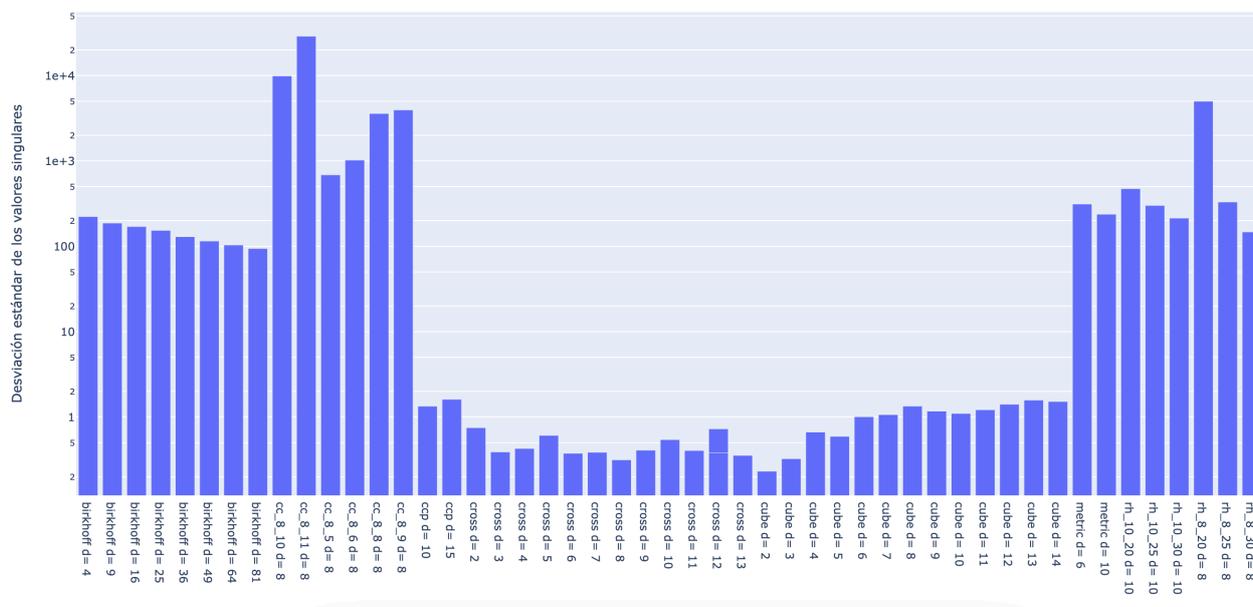


Figura 5.10: Por cada politopo se sortearon 10^6 puntos utilizando RDHR y luego calculamos los valores singulares para el conjunto de puntos sorteados. En la gráfica, se puede ver la desviación estándar de los valores singulares calculados por politopo. Los datos se presentan en escala logarítmica.

Entonces, en el gráfico de la figura 5.13, mostramos la desviación estándar de los valores singulares calculados por cada politopo. Se puede ver que los politopos que tienen mayor desviación estándar, son los que tuvieron mayor error relativo.

Cabe destacar que en los tres algoritmos implementados por la librería VolEsti, antes de hacer el cálculo de volumen, se aplica un proceso de redondeo al polítopo para mejorar el rendimiento. Puede estar pasando, que este redondeo si sea eficiente en el caso de los politopos rh_m_n (que en nuestro caso, incluimos los resultados de las estimaciones obtenidas con estos algoritmos dentro de las estimaciones aceptables), pero no en los politopos cc_m_n , porque estos tienen direcciones en las cuales el politopo es mucho más “largo” que en otras.

5.5. Resultados Modelo compacto E. coli

En esta sección utilizaremos uno de los casos de prueba desarrollados en [34], con el fin de evaluar los resultados obtenidos al aplicar los algoritmos en los modelos metabólicos presentados en 3.3. En particular el Modelo compacto E. coli. Dicho modelo cuenta con 25 flujos, 13 metabolitos y dimensión del espacio de soluciones igual a 12. Por otra parte, este modelo también cuenta con siete reacciones de fermentación (LAC, ETOH, AC, FOR, aKG, GLU, SUCC) que se corresponden con las columnas V08, V09, V10, V11, V12, V13, V17 en la matriz estequiométrica 3.3.

En la sección 5.4.1. Apagando fermentaciones de [34] se procede a hacer el cálculo del volumen del espacio de soluciones en las siguientes situaciones:

- 7 casos prendiendo una de las fermentaciones cada vez
- 21 casos prendiendo dos fermentaciones a la vez.

Cuando se habla de “prender o apagar” una fermentación se refiere a incluir o no en el modelo la columna que la representa.

Debido a los nuevos métodos de cálculo de volumen estudiado, vamos a extender el calculo realizado en dicho trabajo, incrementando la cantidad de fermentaciones prendidas a la vez hasta que todas estén prendidas.

- 35 casos prendiendo tres fermentaciones a la vez.
- 35 casos prendiendo cuatro fermentaciones a la vez.
- 21 casos prendiendo cinco fermentaciones a la vez.
- 7 casos prendiendo seis fermentaciones a la vez.

- 1 caso prendiendo todas las fermentaciones.

La idea es aplicar los algoritmos a estos 126 modelos para poder realizar una comparación entre los resultados de estos métodos y verificar si se podría llegar a aceptar al promedio de dichos resultados como una buena aproximación al volumen.

5.5.1. Comparando volumen exacto con aproximaciones

Antes de aplicar los algoritmos al conjunto de 126 modelos que definimos anteriormente, utilizamos el software Vinci presentado en la sección 2.2, para calcular el volumen exacto de los modelos de menor dimensión (un caso sin ninguna fermentación prendida y 7 casos con una fermentación prendida a la vez). De esta manera, se logró hacer una primera evaluación del porcentaje de error de estimación en cada aproximación obtenida aplicando los algoritmos estudiados.

En la tabla 5.14 se presentan los resultados obtenidos para cada algoritmo junto al porcentaje de error relativo porcentual.

| <i>P</i> | <i>Volumen</i> | <i>CDHR</i> μ | <i>CDHR</i> %Error | <i>FC</i> μ | <i>FC</i> %Error | <i>MCMC</i> μ | <i>MCMC</i> %Error | <i>CB</i> μ | <i>CB</i> %Error | <i>CG</i> μ | <i>CG</i> %Error |
|----------------|----------------|-------------------|--------------------|-----------------|------------------|-------------------|--------------------|-----------------|------------------|-----------------|------------------|
| Todas apagadas | 1.88e+08 | 1.87e+08 | 0.56 | 1.98e+08 | 5.66 | 9.58e+07 | 48.91 | 2.02e+08 | 7.70 | 1.59e+08 | 15.23 |
| V08 | 1.52e+11 | 1.44e+09 | 99.06 | 1.05e+09 | 99.31 | 5.58e+08 | 99.63 | 1.52e+09 | 99.00 | 1.40e+09 | 99.08 |
| V09 | 1.87e+11 | 1.68e+09 | 99.10 | 1.47e+09 | 99.21 | 8.39e+08 | 99.55 | 1.91e+09 | 98.98 | 1.82e+09 | 99.03 |
| V10 | 3.23e+11 | 3.66e+09 | 98.87 | 3.58e+09 | 98.89 | 2.50e+09 | 99.23 | 3.60e+09 | 98.89 | 3.68e+09 | 98.86 |
| V11 | 4.43e+11 | 2.65e+09 | 99.40 | 2.11e+09 | 99.52 | 1.18e+09 | 99.73 | 2.78e+09 | 99.37 | 2.40e+09 | 99.46 |
| V12 | 1.70e+11 | 1.75e+09 | 98.97 | 1.62e+09 | 99.04 | 1.01e+09 | 99.41 | 1.99e+09 | 98.83 | 1.71e+09 | 98.99 |
| V13 | 1.53e+11 | 1.50e+09 | 99.02 | 1.75e+09 | 98.86 | 6.52e+08 | 99.57 | 1.40e+09 | 99.08 | 1.48e+09 | 99.03 |
| V17 | 1.99e+11 | 2.17e+09 | 98.91 | 2.30e+09 | 98.85 | 1.35e+09 | 99.32 | 2.34e+09 | 98.82 | 6.89e+11 | 246.03 |

Tabla 5.14: Resultados modelo compacto E.coli con una fermentación prendida. Restricciones originales.

Se puede ver, que incluso con estos modelos de bajas dimensiones las aproximaciones obtenidas con los cinco algoritmos aplicados resultaron con un error cercano al 100%. Estos resultados fueron obtenidos utilizando $N = 10^6$ en cada algoritmo. En este punto del trabajo, intentamos mejorar los resultados obtenidos utilizando mayores valores de N , en el caso de los algoritmos que fueron implementados en este trabajo y jugando con los parámetros ϵ y N en los algoritmos de la librería VolEsti, pero en ninguno de los dos casos pudimos obtener resultados con un error menor al 97%.

Entendemos que estos malos resultados se deben a la forma de los modelos metabólicos que estamos utilizando (igual que pasó con los modelos *cc_m_n* en la sección 5.4.2). Estos espacios de soluciones tienen forma alargada para las reacciones que no están muy restringidas, y angosta para los flujos más restringidos, lo que provoca que una gran parte del espacio de soluciones se encuentre cercana al borde del mismo. Esto hace que las técnicas de muestreo aleatorio utilizadas en todos los algoritmos estudiados no funcionen bien porque la distancia que se puede recorrer a partir de los puntos muestreados en la mayoría de las direcciones tiende a ser muy pequeña. Esto provoca que todos los puntos estén muy cercanos entre sí, haciendo que no se tenga una buena generación de un muestreo uniformemente distribuido.

De todas maneras, cabe destacar que en el caso de los algoritmos CG y CB implementados en VolEsti, existe un paso previo a la aproximación del volumen, que realiza una transformación lineal sobre el modelo. Dicha transformación resulta en un espacio de soluciones más “redondeado” a partir del cual es posible estimar el volumen del espacio de soluciones original. Esta etapa de “redondeo” se realiza justamente para evitar los problemas de muestreo aleatorio ocasionados por la forma alargada de los modelos. De todas maneras, en estos ejemplos vemos que incluso con esta técnica de redondeo no fue posible obtener buenos resultados.

Por otra parte, existen métodos de muestreo de flujos metabólicos que abordan esta problemática. En [34] se utilizaron los métodos de muestreo de flujos metabólicos Artificial Centering Hit-and-Run (ACHR) y optGp-Samoler, proporcionados por COBRAPy¹, que es un paquete que contiene herramientas de análisis de modelos metabólicos y procesos biológicos. Ambos métodos se basan en el algoritmo HR, pero están adaptados al caso particular de muestreos dentro de espacios de soluciones en estado estacionario de redes metabólicas (espacios de soluciones muy alargados en algunas direcciones con respecto a otras). Para evitar este problema estos algoritmos intentan hacer los muestreos en las direcciones más largas. Tal vez, combinando estas técnicas de muestreo con los algoritmos específicamente desarrolladas para este tipo de modelos con los algoritmos CG y CB, se logre mejorar los resultados obtenidos para este tipo de modelos.

¹Constraints-Based Reconstruction and Analysis for Python. <https://cobrapy.readthedocs.io/en/latest/>

5.5.2. Ajuste del modelo metabólico.

A continuación recordaremos brevemente como está definido el modelo metabólico compacto. El mismo se compone de la matriz estequiométrica en donde cada columna representa una reacción y cada fila representa un metabolito. Por otra parte, las reacciones se encuentran restringidas por los límites superior e inferior, como puede verse en 5.15.

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 | V22 | V23 | V24 | V25 | VBM |
|-------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|-----|-----|--------|
| ACoA | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | -1 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -3.748 |
| ADP | -1 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | -1.25 | 0 | 0 | 59.81 |
| aKG | 0 | 0 | 0 | 1 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.1182 |
| ATP | 1 | 1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | 1.25 | 0 | 0 | -59.81 |
| FUM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 |
| GLN | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.256 |
| GLU | 0 | 0 | 0 | 0 | 1 | -1 | 2 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -4.941 |
| NAD | -2 | 0 | -1 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 0 | 0 | 0 | -1 | 0 | 1 | 0 | 1 | -3.547 |
| NADH | 2 | 0 | 1 | 0 | 0 | 0 | 0 | -1 | -2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | -1 | 3.547 |
| NADP | 0 | 0 | 0 | -1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 13.028 |
| NADPH | 0 | 0 | 0 | 1 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | -13.03 |
| OAA | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1.787 |
| PEP | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.519 |
| PYR | 1 | 1 | -1 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -2.833 |
| q8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | -1 | 0 |
| q8h2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | -1 | 1 | 0 |
| SUCC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| | LB | UB |
|-----|-------|------|
| V1 | 0 | 10 |
| V2 | 0 | 1000 |
| V3 | 0 | 1000 |
| V4 | 0 | 1000 |
| V5 | -1000 | 1000 |
| V6 | 0 | 1000 |
| V7 | 0 | 1000 |
| V8 | 0 | 1000 |
| V9 | 0 | 1000 |
| V10 | 0 | 1000 |
| V11 | 0 | 1000 |
| V12 | 0 | 1000 |
| V13 | 0 | 1000 |
| V14 | 0 | 1000 |
| V15 | 0 | 1000 |
| V16 | -1000 | 1000 |
| V17 | 0 | 1000 |
| V18 | 0 | 1000 |
| V19 | 8.39 | 1000 |
| V20 | -1000 | 1000 |
| V22 | 0 | 1000 |
| V23 | 0 | 1000 |
| V24 | 0 | 1000 |
| V25 | 0 | 1000 |

Tabla 5.15: Modelo compacto E. coli. Matriz estequiométrica y Restricciones

En este modelo, las restricciones con valor -1000 o 1000 corresponden a reacciones biológicas que en realidad no tienen restricción. Tener estas restricciones con estos valores es lo que hace que los algoritmos no funcionen bien por todos los motivos explicados anteriormente. Como este valor es un valor arbitrario, y el modelo es una representación simplificada de la realidad, decidimos ajustar el modelo y cambiar las restricciones que están en 1000 a 500 para ver si con esto podemos mejorar los resultados de aproximación obtenidos por los algoritmos estudiados. El nuevo conjunto de restricciones pasa a ser el de la tabla 5.16

| | LB | UB |
|-----|------|-----|
| V1 | 0 | 10 |
| V2 | 0 | 500 |
| V3 | 0 | 500 |
| V4 | 0 | 500 |
| V5 | -500 | 500 |
| V6 | 0 | 500 |
| V7 | 0 | 500 |
| V8 | 0 | 500 |
| V9 | 0 | 500 |
| V10 | 0 | 500 |
| V11 | 0 | 500 |
| V12 | 0 | 500 |
| V13 | 0 | 500 |
| V14 | 0 | 500 |
| V15 | 0 | 500 |
| V16 | -500 | 500 |
| V17 | 0 | 500 |
| V18 | 0 | 500 |
| V19 | 8.39 | 500 |
| V20 | -500 | 500 |
| V22 | 0 | 500 |
| V23 | 0 | 500 |
| V24 | 0 | 500 |
| V25 | 0 | 500 |

Tabla 5.16: Modelo compacto E. coli. Restricciones modificadas

Cabe destacar, que a pesar de que no contamos con los conocimientos como para entender a nivel biológico el impacto que puede llegar a tener este cambio en las restricciones de los modelos, sabemos que la red de E. coli que es utilizada en [26] para validar la utilidad del método basado en MCMC, es presentada como ejemplo de los análisis a nivel biológico que se pueden llegar a hacer al poder calcular el volumen del espacio de solución de estos modelos. Dicha red metabólica, consta de 92 reacciones y 76 metabolitos. El politopo correspondiente tiene una dimensión de $d = 24$ y los flujos están restringidos por $v_{min} = 10 \leq v \leq 10 = v_{max}$. Por este motivo, es que entendemos que aun después de hacer el cambio en el valor de las restricciones (de 1000 a 500) seguiría siendo posible realizar un análisis desde el punto de vista biológico del comportamiento del microorganismo a partir del modelo ajustado.

Finalmente, luego de modificar las restricciones del modelo volvimos a calcular el volumen exacto y a aplicar los algoritmos para aproximar dicho volumen. En la tabla 5.17 se pueden ver los resultados obtenidos en este caso.

| P | Volumen | CDHR μ | CDHR %Error | FC μ | FC %Error | MCMC μ | MCMC %Error | CB μ | CB %Error | CG μ | CG %Error |
|----------------|----------|------------|-------------|----------|-----------|------------|-------------|----------|-----------|----------|-----------|
| Todas apagadas | 1.88e+08 | 1.90e+08 | 1.16 | 1.93e+08 | 2.75 | 1.08e+08 | 42.51 | 1.85e+08 | 1.31 | 1.91e+08 | 1.95 |
| V08 | 1.38e+09 | 1.32e+09 | 4.74 | 1.43e+09 | 3.59 | 1.28e+10 | 829.13 | 1.44e+09 | 4.00 | 1.45e+09 | 4.97 |
| V09 | 1.70e+09 | 1.78e+09 | 4.69 | 1.68e+09 | 1.52 | 4.96e+10 | 2817.18 | 1.71e+09 | 0.47 | 1.69e+09 | 0.64 |
| V10 | 3.54e+09 | 3.46e+09 | 2.22 | 2.98e+09 | 15.82 | 3.35e+10 | 847.61 | 3.52e+09 | 0.56 | 3.76e+09 | 6.19 |
| V11 | 2.59e+09 | 2.49e+09 | 3.70 | 2.21e+09 | 14.70 | 8.93e+10 | 3351.60 | 2.49e+09 | 3.63 | 2.61e+09 | 0.97 |
| V12 | 1.75e+09 | 1.72e+09 | 1.40 | 1.59e+09 | 8.98 | 1.99e+10 | 1040.78 | 1.74e+09 | 0.26 | 1.74e+09 | 0.26 |
| V13 | 1.51e+09 | 1.47e+09 | 2.92 | 1.66e+09 | 9.37 | 8.40e+09 | 455.24 | 1.57e+09 | 3.80 | 1.41e+09 | 6.55 |
| V17 | 2.18e+09 | 2.11e+09 | 2.95 | 1.83e+09 | 15.95 | 1.65e+10 | 659.49 | 2.22e+09 | 2.23 | 2.05e+09 | 5.74 |
| V08V09 | 8.59e+09 | 8.49e+09 | 1.10 | 1.15e+10 | 33.92 | 1.85e+10 | 115.10 | 8.64e+09 | 0.55 | 9.60e+09 | 11.79 |
| V08V10 | 1.95e+10 | 1.71e+10 | 12.77 | 1.82e+10 | 6.82 | 4.15e+10 | 112.21 | 1.70e+10 | 12.92 | 1.81e+10 | 7.41 |
| V08V11 | 1.59e+10 | 1.65e+10 | 3.61 | 1.00e+10 | 37.00 | 6.73e+10 | 323.55 | 1.56e+10 | 1.95 | 1.50e+10 | 5.72 |
| V08V12 | 8.96e+09 | 8.83e+09 | 1.46 | 7.23e+09 | 19.25 | 3.44e+10 | 283.61 | 9.21e+09 | 2.82 | 1.30e+10 | 45.64 |
| V08V13 | 7.81e+09 | 8.09e+09 | 3.59 | 6.80e+09 | 12.86 | 1.79e+10 | 129.58 | 8.07e+09 | 3.37 | 8.43e+09 | 7.94 |
| V08V17 | 1.15e+10 | 1.03e+10 | 10.40 | 2.07e+10 | 80.39 | 3.69e+10 | 221.71 | 1.16e+10 | 1.29 | 1.15e+10 | 0.61 |
| V09V10 | 1.79e+10 | 1.91e+10 | 6.55 | 1.77e+10 | 1.28 | 6.37e+10 | 256.12 | 1.91e+10 | 6.64 | 1.94e+10 | 8.19 |
| V09V11 | 2.28e+10 | 2.22e+10 | 2.33 | 1.87e+10 | 17.81 | 2.06e+11 | 806.53 | 2.40e+10 | 5.33 | 2.31e+10 | 1.62 |
| V09V12 | 1.15e+10 | 1.06e+10 | 8.01 | 8.69e+09 | 24.50 | 7.84e+10 | 581.33 | 1.09e+10 | 4.97 | 1.18e+10 | 2.90 |
| V09V13 | 1.00e+10 | 1.01e+10 | 0.23 | 2.23e+10 | 122.29 | 2.52e+10 | 150.90 | 9.65e+09 | 3.97 | 9.54e+09 | 5.03 |
| V09V17 | 1.43e+10 | 1.46e+10 | 2.02 | 1.12e+10 | 21.52 | 1.10e+11 | 667.83 | 1.45e+10 | 1.62 | 1.46e+10 | 2.32 |
| V10V11 | 4.92e+10 | 5.04e+10 | 2.50 | 4.80e+10 | 2.29 | 2.63e+11 | 434.19 | 5.19e+10 | 5.47 | 5.19e+10 | 5.63 |
| V10V12 | 1.96e+10 | 1.94e+10 | 0.93 | 1.83e+10 | 6.36 | 9.62e+10 | 391.21 | 1.93e+10 | 1.31 | 1.97e+10 | 0.79 |
| V10V13 | 1.71e+10 | 1.72e+10 | 0.78 | 1.38e+10 | 19.05 | 6.61e+10 | 286.73 | 1.72e+10 | 0.38 | 1.79e+10 | 4.41 |
| V10V17 | 2.67e+10 | 2.64e+10 | 0.91 | 3.50e+10 | 30.99 | 1.15e+11 | 330.56 | 2.86e+10 | 7.00 | 2.84e+10 | 6.42 |
| V11V12 | 2.14e+10 | 2.11e+10 | 1.66 | 2.55e+10 | 18.89 | 1.45e+11 | 575.94 | 2.29e+10 | 6.57 | 2.17e+10 | 1.13 |
| V11V13 | 1.90e+10 | 1.87e+10 | 1.54 | 1.72e+10 | 9.52 | 1.36e+11 | 616.94 | 1.92e+10 | 1.09 | 1.78e+10 | 6.24 |
| V11V17 | 2.68e+10 | 2.75e+10 | 2.60 | 2.27e+10 | 15.46 | 3.89e+11 | 1350.28 | 2.71e+10 | 1.05 | 2.73e+10 | 1.76 |
| V12V13 | 1.11e+10 | 7.16e+09 | 35.46 | 6.77e+09 | 38.93 | 1.82e+10 | 64.40 | 7.47e+09 | 32.65 | 7.33e+09 | 33.90 |
| V12V17 | 9.64e+09 | 9.37e+09 | 2.73 | 8.44e+09 | 12.46 | 1.75e+10 | 81.62 | 9.99e+09 | 3.65 | 9.49e+09 | 1.54 |
| V13V17 | 9.09e+09 | 8.99e+09 | 1.12 | 1.11e+10 | 21.76 | 3.48e+10 | 283.36 | 9.07e+09 | 0.19 | 8.82e+09 | 2.95 |

Tabla 5.17: Resultados para el modelo compacto E.coli con una y dos fermentaciones prendidas.

A partir de estos resultados, decidimos seleccionar los algoritmos CG y CB (porque fueron los que funcionaron mejor con un menor porcentaje de error y en general tuvieron un menor tiempo de ejecución) para ser aplicados al conjunto completo de 126 modelos que definimos anteriormente. Luego de obtener los 126 resultados aplicamos el método Bland-Altman [16] para determinar si podemos considerar que ambos métodos (CG y CB) están convergiendo a valores similares. Si podemos confirmar este comportamiento mediante este análisis estadístico, podríamos considerarlo como un indicador de que nos estamos acercando al volumen real del politopo, de lo contrario, el análisis nos puede ayudar a descartar resultados inválidos.

El método Bland-Altman (también conocido como Gráfico de diferencia de medias de Tukey) es utilizado para determinar si 2 métodos diferentes que miden la misma propiedad concuerdan lo suficiente para que se pueda decir que son intercambiables. Aunque el método en sí mismo no responde esta pregunta, lo que hace es cuantificar el sesgo y un rango de concordancia, dentro del cual se incluyen el 95 % de las diferencias entre una medición y la otra. Corresponde al problema puntual de medición determinar si estas diferencias son suficientemente pequeñas como para considerar que los dos métodos son intercambiables o no. En nuestro caso, como no conocemos el valor verdadero y ninguno de los dos métodos que tenemos es una referencia de dicho valor, vamos a usar este método para determinar si con este tipo de modelos, podemos decir que ambos algoritmos convergen a valores similares.

Los principales aspectos que considera este método son:

- Como concuerdan los métodos en promedio
- Como concuerdan los métodos para cada caso particular.

Para esto se calcula, para cada medición, la diferencia entre las medidas obtenidas con los dos métodos (en nuestro caso, $d = CG - CB$). La media de esta diferencia μ_d representa el error sistemático, mientras que la varianza σ_d^2 de estas diferencias mide la dispersión del error aleatorio. A partir de estas dos medidas si la distribución de las diferencias es la normal (si las diferencias no se distribuyen normalmente, se puede intentar una transformación logarítmica de los datos originales), se calculan los límites de concordancia del 95 % como $\mu_d \pm 1,96 \cdot \sigma_d$. Estos límites indican entre qué diferencias oscilan la mayor parte de las medidas tomadas con los 2 métodos.

Finalmente, se grafica la diferencia entre los resultados de cada aproximación ($d = CG - CB$) en función de la media entre ambos resultados ($(CG + CB)/2$). Graficar la diferencia contra la media también permite investigar cualquier posible relación entre el error de medición y la media de ambas mediciones (por ejemplo que a medida que aumenta la media también aumenta el error).

Por otra parte, es importante evaluar las diferencias en diferentes magnitudes de la variable medida, por eso es que realizamos este análisis sobre los 126 politopos construidos a partir de las diferentes combinaciones de fermentaciones prendidas en el modelo metabólico.

En el anexo G.4.1 se encuentran las tabla con todos los resultados obtenidos.

5.5.3. Análisis de resultados

En esta sección la idea principal es analizar si a partir de los resultados obtenidos por ambos métodos podemos decir con un determinado nivel de probabilidad que ambos algoritmos están convergiendo a valores similares (en el caso ideal, el volumen exacto del politopo). Para esto, como se comentó en la sección anterior, vamos a aplicar el método de Bland-Altman para determinar los límites dentro de los cuales se espera que se encuentren el 95 % de las diferencias entre los resultados de ambos métodos.

Antes de comenzar con el análisis mencionado, vamos a graficar los valores obtenidos por cada método, uno en función del otro. De esta manera, se puede tener una primera impresión sobre la relación entre los resultados. Si ambos métodos nos permitieran obtener el volumen exacto de los politopos, esta gráfica debería coincidir exactamente con la gráfica de la función $x = y$.

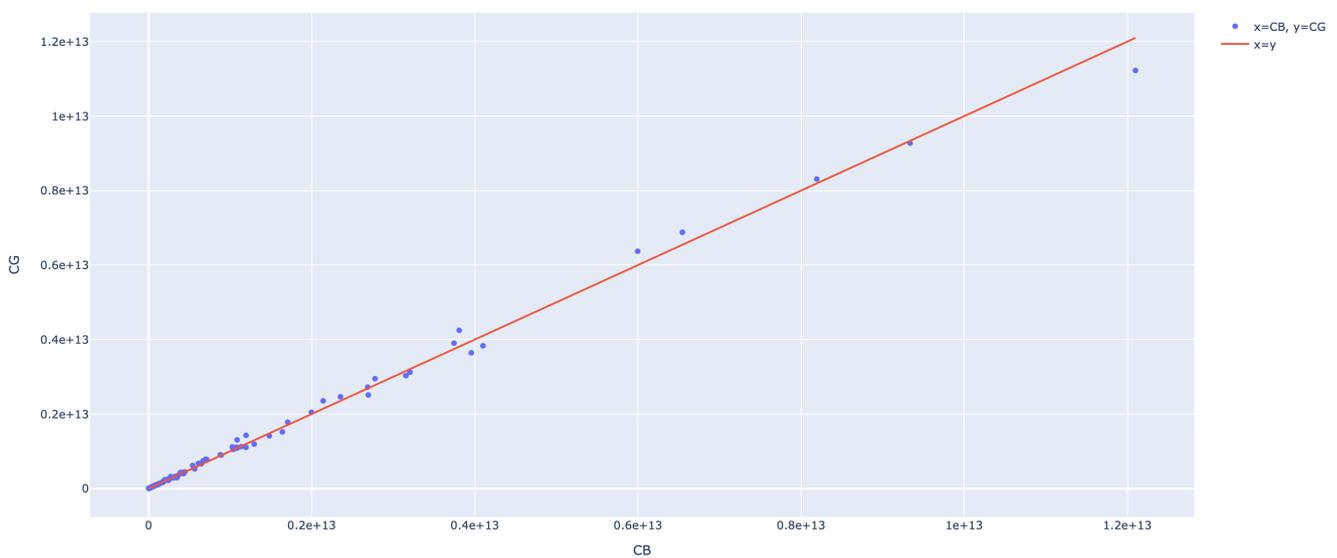


Figura 5.11: Gráfica del volumen aproximado del modelo compacto E.coli , calculado con el algoritmo CB en función del volumen calculado con CG.

A simple vista podemos decir que los valores obtenidos se acercan bastante a la igualdad, aunque a medida que aumenta el volumen vemos que los puntos se alejan más de la gráfica $x = y$ de referencia.

A continuación, para saber si podemos aplicar el método Bland-Altman, verificamos si el conjunto de diferencias obtenido siguen la distribución normal. Para esto, primero aplicaremos un conjunto de test de hipótesis, disponibles en la librería SciPy de Python, que permiten afirmar que es muy probable que los datos obtenidos sigan la distribución normal dada la evidencia disponible.

Cada prueba calcula una estadística específica. Esta estadística puede ayudar en la interpretación del resultado, aunque esto requiere un conocimiento más profundo de la prueba estadística específica. En cambio, el valor p se puede utilizar para interpretar la estadística de forma rápida y precisa en aplicaciones prácticas.

La hipótesis nula de las pruebas H_0 es suponer que la muestra se extrajo de una distribución gaussiana. Se elige el nivel de significancia de la prueba α , típicamente 5 % (o 0.05), que se usa para interpretar el valor p .

El valor de p se interpreta de la siguiente manera:

- $p \leq \alpha$: rechazar H_0 , no es normal.
- $p > \alpha$: falla al rechazar H_0 , normal.

Un resultado superior al 5 % no significa que la hipótesis nula sea cierta. Significa que es muy probable que sea cierta dada la evidencia disponible.

Las tres pruebas cuya implementación se encuentra disponible en SciPy que utilizamos son: prueba de Shapiro-Wilk, prueba de Anderson-Darling, Prueba de K^2 de D'Agostino. Las pruebas Shapiro-Wilk y K^2 de D'Agostino devuelven la estadística y el valor p mientras que la prueba de Anderson-Darling devuelve una lista de valores críticos en lugar de un solo valor p .

En las tablas 5.18 y 5.19 se muestran los resultados de los test realizados. Como se puede ver, el conjunto de las diferencias no paso las pruebas de normalidad, por lo que no vamos a poder aplicar el análisis al conjunto $d = CG - CB$.

| Test | Cantidad de muestras | α | Estadístico | p-value | Resultado |
|------------------|----------------------|----------|-------------|--------------|------------------|
| Shapiro-Wilk | 126 | 0.05 | 0.583612 | 2.584454e-17 | Se rechaza H_0 |
| K^2 D'Agostino | 126 | 0.05 | 112.343118 | 4.027187e-25 | Se rechaza H_0 |

Tabla 5.18: Resultados pruebas Shapiro-Wilk y K^2 de D'Agostino para verificar normalidad en el conjunto de diferencias.

| Cantidad de muestras | α | Estadístico | Valor crítico | Resultado |
|----------------------|----------|-------------|---------------|------------------|
| 126 | 0.15 | 17.999 | 0.559 | Se rechaza H_0 |
| 126 | 0.10 | 17.999 | 0.637 | Se rechaza H_0 |
| 126 | 0.05 | 17.999 | 0.764 | Se rechaza H_0 |
| 126 | 0.025 | 17.999 | 0.891 | Se rechaza H_0 |
| 126 | 0.01 | 17.999 | 1.06 | Se rechaza H_0 |

Tabla 5.19: Resultados pruebas Anderson-Darling para verificar normalidad en el conjunto de diferencias.

Procedemos entonces a realizar los mismos test pero sobre el conjunto $d = \ln(CG) - \ln(CB)$. Si se cumple que dicha transformación del conjunto de diferencias registradas entre ambos métodos sigue la distribución normal, estaremos en condiciones de decir que con un alto grado de probabilidad la relación $\frac{CG}{CB}$ se encuentra acotada por los siguientes valores:

$$lb < \ln(CG) - \ln(CB) < ub \rightarrow e^{lb} < e^{(\ln(CG) - \ln(CB))} < e^{ub} \rightarrow e^{lb} < e^{(\ln(\frac{CG}{CB}))} < e^{ub} \rightarrow e^{lb} < \frac{CG}{CB} < e^{ub} \quad (5.1)$$

En donde $lb = \mu_d - 1,96 \cdot \sigma_d$ y $ub = \mu_d + 1,96 \cdot \sigma_d$.

Y efectivamente, según los resultados mostrados en las tablas 5.20 y 5.21, se puede decir que es muy probable que $d = \ln(CG) - \ln(CB)$ siga la distribución normal.

| Test | Cantidad de muestras | α | Estadístico | p-value | Resultado |
|------------------|----------------------|----------|-------------|----------|----------------------------|
| Shapiro-Wilk | 126 | 0.05 | 0.983070 | 0.117223 | No se puede rechazar H_0 |
| K^2 D'Agostino | 126 | 0.05 | 2.457571 | 0.292647 | No se puede rechazar H_0 |

Tabla 5.20: Resultados pruebas Shapiro-Wilk y K^2 de D'Agostino para verificar normalidad en el conjunto de diferencias.

| Cantidad de muestras | α | Estadístico | Valor crítico | Resultado |
|----------------------|----------|-------------|---------------|----------------------------|
| 126 | 0.15 | 0.5979 | 0.559 | No se puede rechazar H_0 |
| 126 | 0.10 | 0.5979 | 0.637 | No se puede rechazar H_0 |
| 126 | 0.05 | 0.5979 | 0.764 | No se puede rechazar H_0 |
| 126 | 0.025 | 0.5979 | 0.891 | No se puede rechazar H_0 |
| 126 | 0.01 | 0.5979 | 1.06 | No se puede rechazar H_0 |

Tabla 5.21: Resultados pruebas Anderson-Darling para verificar normalidad en el conjunto de diferencias.

Por último, realizamos el gráfico de $d = \ln(CG) - \ln(CB)$ en función de $\bar{d} = (\ln(CG) + \ln(CB))/2$, y calculamos los límites de concordancia $lb = \mu_d - 1,96 \cdot \sigma_d$ y $ub = \mu_d + 1,96 \cdot \sigma_d$.

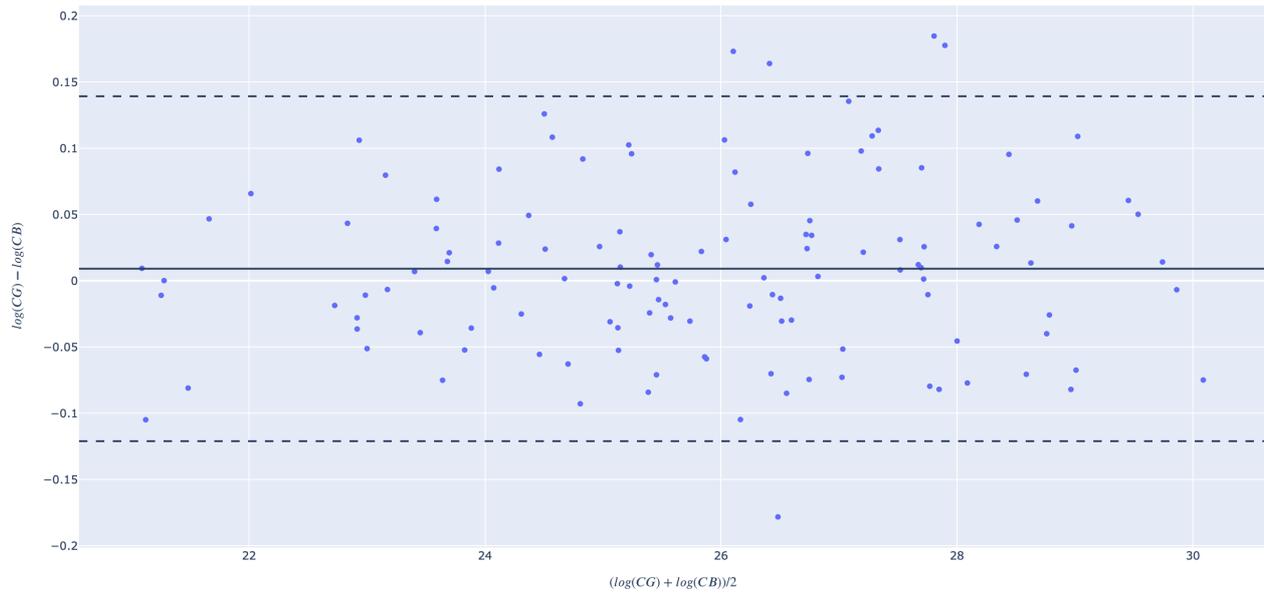


Figura 5.12: Gráfico de $d = \ln(CG) - \ln(CB)$ en función de $\bar{d} = (\ln(CG) + \ln(CB))/2$, la línea constante representa el valor medio del conjunto de diferencias d y las líneas punteadas son límites de concordancia $lb = \mu_d - 1,96 \cdot \sigma_d$ y $ub = \mu_d + 1,96 \cdot \sigma_d$.

$$\mu_d = 0,008979, \sigma_d = 0,066417, lb = -0,121198, ub = 0,139156 \quad (5.2)$$

Finalmente, considerando 5.1 y 5.2 podemos decir que con un 95% de probabilidad, cuando utilizamos los algoritmos CG y CB para aproximar el volumen de los modelos metabólicos que tienen similares características a los generados a partir del modelo compacto E.coli, se cumple que:

$$0,885858 < \frac{CG}{CB} < 1,149303$$

Lo que indica que en general, ambos algoritmos están retornando estimaciones similares.

En resumen, a lo largo de esta sección generamos 126 casos de prueba a partir del modelo metabólico de E. coli. Dentro de esos 126 casos, hubo un subconjunto para el cual pudimos calcular el volumen exacto utilizando el software Vinci. Luego de tener este volumen calculado pudimos evaluar el porcentaje de error que tuvo cada método de aproximación de volumen estudiado en la sección anterior. A partir de esta primera evaluación, seleccionamos los dos métodos que tuvieron menor porcentaje de error (CG y CB). Una vez que definimos qué métodos se iban a utilizar para los siguientes pasos, aplicamos los mismos al conjunto entero de 126 modelos. Esto nos permitió generar una muestra de resultados con magnitudes de volúmenes heterogéneas y de un tamaño considerable. A partir de esta muestra pudimos aplicar el método de Bland-Altman que es utilizado para determinar si 2 métodos diferentes que miden la misma propiedad concuerdan lo suficiente para que se pueda decir que son intercambiables. Con este método estadístico, pudimos obtener una cota a la relación entre los resultados de ambos algoritmos. Dicha cota nos indica que en los casos que estamos estudiando los dos métodos de aproximación de volumen están convergiendo a valores similares. Es claro que la precisión de la estimación del volumen que se quiere llegar a tener dependerá del problema biológico que se busque analizar a través del cálculo del volumen de estos modelos, pero a los efectos de este trabajo, si consideramos los porcentajes de error calculado para los modelos de dimensiones más bajas y el hecho de que ambos algoritmos convergen a valores similares, podemos considerar que los resultados obtenidos aplicando los algoritmos CG y CB en el modelo compacto E.coli son aceptables. Aunque es importante recordar que para tener estos resultados tuvimos que hacer ajustes al modelo, es decir que no pudimos obtener los resultados deseados con el modelo original planteado en [34]

5.6. Resultados Modelo núcleo E. coli

En esta sección intentamos repetir el mismo procedimiento que en la sección anterior utilizando como base para generar los 126 modelos al modelo núcleo de E.coli. Como el modelo compacto, fue construido en [34] a

partir del modelo núcleo, tenemos que cada columna que representa una fermentación en el modelo compacto, se corresponde con una o más columnas en el modelo núcleo. Teniendo las siguientes relaciones:

- **V08:** columnas 60 y 16 en modelo núcleo.
- **V09:** columnas 0, 9 y 18 en modelo núcleo.
- **V10:** columnas 81, 2 y 5 en modelo núcleo.
- **V11:** columnas 72 y 42 en modelo núcleo.
- **V12:** columna 8 en modelo núcleo
- **V13:** columna 55 en modelo núcleo.
- **V17:** columna 87 en modelo núcleo.

En este caso, se repiten los mismos experimentos que en el modelo compacto, con la diferencia de que cuando se habla de prender o apagar una fermentación, se habla de incluir o eliminar el conjunto de columnas que la representan en el modelo. Igual que en el caso anterior, ajustamos las restricciones del modelo pasando de -1000 a -500 y 1000 a 500 los valores mínimos y máximos respectivamente.

En el anexo G.4.2 se pueden ver los resultados obtenidos.

5.6.1. Análisis de resultados

Graficando los valores obtenidos por cada método (CG en función de CB) podemos ver que, a diferencia del modelo compacto, en este caso para los politopos con volumen más grande la diferencia entre los resultados de ambos métodos es incluso más grande que el orden de magnitud de los resultados. A simple vista, parece que los resultados obtenidos no pueden ser considerados como válidos.

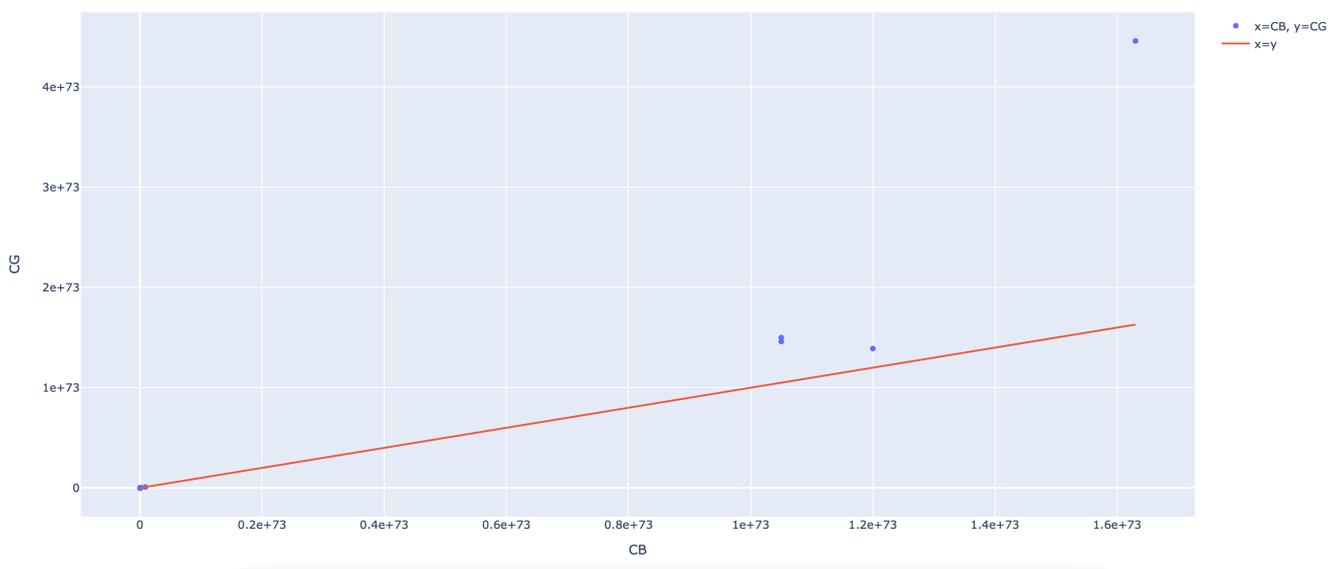


Figura 5.13: Gráfica del volumen aproximado del modelo núcleo E.coli , calculado con el algoritmo CB en función del volumen calculado con CG.

De todas maneras procedimos a realizar las pruebas de normalidad al conjunto $d = \ln(CG) - \ln(CB)$. Como se puede ver en las tablas 5.22 y 5.23 la conclusión de las pruebas es que este conjunto de datos no sigue la distribución normal.

| Test | Cantidad de muestras | α | Estadístico | p-value | Resultado |
|------------------|----------------------|----------|-------------|--------------|------------------|
| Shapiro-Wilk | 126 | 0.05 | 0.555044 | 7.201473e-18 | Se rechaza H_0 |
| K^2 D'Agostino | 126 | 0.05 | 141.786896 | 1.626922e-31 | Se rechaza H_0 |

Tabla 5.22: Resultados pruebas Shapiro-Wilk y K^2 de D'Agostino para verificar normalidad en el conjunto de diferencias.

| Cantidad de muestras | α | Estadístico | Valor crítico | Resultado |
|----------------------|----------|-------------|---------------|------------------|
| 126 | 0.15 | 14.6928 | 0.559 | Se rechaza H_0 |
| 126 | 0.10 | 14.6928 | 0.637 | Se rechaza H_0 |
| 126 | 0.05 | 14.6928 | 0.764 | Se rechaza H_0 |
| 126 | 0.025 | 14.6928 | 0.891 | Se rechaza H_0 |
| 126 | 0.01 | 14.6928 | 1.06 | Se rechaza H_0 |

Tabla 5.23: Resultados pruebas Anderson-Darling para verificar normalidad en el conjunto de diferencias.

Debido a estos resultados no es posible establecer los límites de concordancia definidos por el método Bland-Altman.

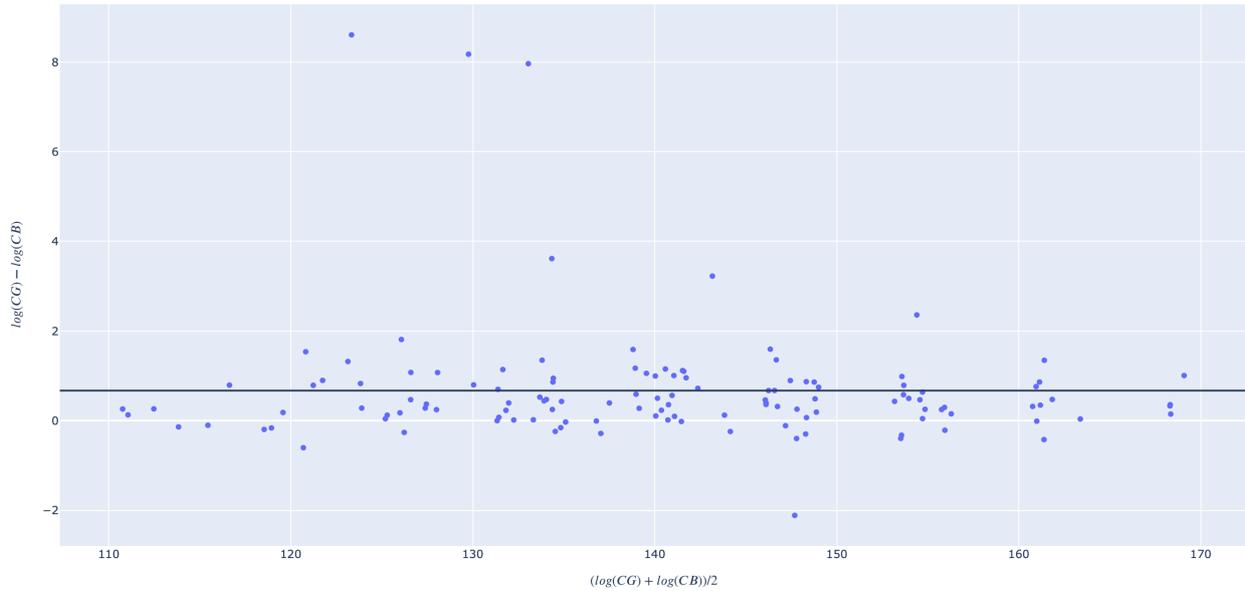


Figura 5.14: Gráfico de $d = \ln(CG) - \ln(CB)$ en función de $\bar{d} = (\ln(CG) + \ln(CB))/2$, la línea constante representa el valor medio del conjunto de diferencias d .

Debido a estos resultados no es posible establecer los límites de concordancia definidos por el método Bland-Altman. Igualmente si graficamos los datos aplicando dicho método (aunque no contemos con los límites de concordancia) vemos como el promedio de de los valores del conjunto d es un valor cercano a dos, si los resultados fueran mejores este promedio debería estar cercano a cero (como en el caso del modelo compacto).

Capítulo 6

Conclusiones

En el presente trabajo nos centramos en el estudio de algoritmos basados en Monte Carlo que pudieran utilizarse para aproximar el volumen de politopos expresados como un conjunto de inecuaciones lineales con el objetivo de aplicar estos algoritmos en los modelos metabólicos compacto y núcleo de *E.coli*. A partir del estudio realizado, encontramos un grupo de algoritmos de Monte Carlo que probamos en un conjunto de politopos recopilado de diferentes trabajos. Los resultados sobre este conjunto fueron buenos en la mayoría de los casos, incluso en los de dimensión 81, que fue la más grande que probamos. Pero al mismo tiempo fueron muy malos para un subconjunto de estos politopos. Después de analizar este subconjunto vimos que eran politopos que tienen forma alargada. Entendemos que los malos resultados obtenidos al aplicar los algoritmos se dan por la dificultad de generar puntos uniformemente distribuidos en las direcciones alargadas usando los métodos de caminatas aleatorias en las que se basan todos estos algoritmos. Esta forma alargada característica es un problema general en los modelos metabólicos, porque la forma de representar la realidad biológica es restringiendo mucho determinadas direcciones cuando los procesos cuentan con alguna limitación física y biológica y configurando las demás restricciones con valores considerablemente más grandes.

Para abordar este problema para otras metodologías de análisis sobre los modelos metabólicos, existen métodos de muestreo de flujos metabólicos que se adaptan a dichas características. En [37] se utilizaron los métodos de muestreo de flujos metabólicos Artificial Centering Hit-and-Run (ACHR) y optGp- Samoler, proporcionados por COBRAPy. Ambos métodos se basan en el algoritmo HR, pero están adaptados al caso particular de muestreos dentro de espacios de soluciones en estado estacionario de redes metabólicas. Para evitar este problema estos algoritmos intentan hacer los muestreos en direcciones más largas. Otra posibilidad para mejorar los resultados que no fue explorada en este trabajo, es el hecho de que todos los algoritmos estudiados permiten algún tipo de paralelización. Por ejemplo, en el método basado en MCMC es posible realizar la ejecución de las diferentes cadenas de Markov de forma paralela. Incluso en el caso de los métodos implementados en la librería VolEsti se cuenta la opción de ingresar la cantidad de hilos con la cual se quiere trabajar para realizar los cálculos de forma paralela.

De todos modos, a pesar de estas dificultades, intentamos aplicar los algoritmos a los modelos compacto y núcleo *E.coli* y en un principio, no pudimos conseguir resultados aceptables. Empezamos haciendo las pruebas con el modelo compacto, pero luego de realizar ajustes para atenuar el problema de la forma alargada, pudimos aplicar el algoritmo. Una vez que verificamos cuales eran los algoritmos que funcionan mejor con esta nueva versión del modelo compacto, generamos un conjunto de modelos basados en un caso de prueba de interés biológico que nos permitió realizar un análisis estadístico de los resultados obtenidos. A partir de este análisis concluimos que para el modelo compacto si pudimos obtener una estimación aceptable a los efectos de este trabajo. Finalmente, cuando pasamos a realizar el mismo análisis al modelo núcleo que tiene dimensión más grande que el anterior, no pudimos llegar a la misma conclusión. Con estos resultados reafirmamos la noción previa que teníamos de que lo que puede afectar el desempeño de los algoritmos estudiados es una combinación entre las características que aportan las restricciones al modelo y la dimensión del mismo.

En resumen, se pudo comprobar que, como dice la literatura consultada, los métodos estudiados en ciertas condiciones permiten obtener una buena aproximación del volumen de H-politopos de dimensiones cercanas a 100. Aunque en el caso de los modelos metabólicos se presentaron dificultades para aproximar el volumen mediante dichos métodos debido a las características particulares de este tipo de modelos. Por otra parte, sabemos que existen adaptaciones de los métodos estudiados para generar puntos uniformemente distribuidos dentro de un espacio de soluciones que abordan la problemática sobre la forma característica de los modelos metabólicos. Se podría intentar combinar estas adaptaciones de caminatas aleatorias con los métodos estudiados y ver si esto permite obtener mejores resultados en modelos sencillos que permiten estudiar las redes metabólicas como los vistos en este trabajo.

Capítulo 7

Bibliografía

- [1] Matthias Beck y Dennis Pixton. “The Ehrhart Polynomial of the Birkhoff Polytope”. En: *Discrete and Computational Geometry* 30.4 (oct. de 2003), págs. 623-637. ISSN: 1432-0444. DOI: 10.1007/s00454-003-2850-8. URL: <http://dx.doi.org/10.1007/s00454-003-2850-8>.
- [2] Arne Brøndsted. *An Introduction to Convex Polytopes*. Graduate Texts in Mathematics. Springer New York, 2012.
- [3] Benno Büeler y Andreas Enge. *Software VINCI*. <http://www.multiprecision.org/vinci/home.html>. Accedido 08-08-2021.
- [4] Apostolos Chalkis, Ioannis Z. Emiris y Vissarion Fisikopoulos. *Practical Volume Estimation by a New Annealing Schedule for Cooling Convex Bodies*. 2019. arXiv: 1905.05494 [cs.CG].
- [5] Apostolos Chalkis y Vissarion Fisikopoulos. *volesti: Volume Approximation and Sampling for Convex Polytopes in R*. Inf. téc. 2020. arXiv: 2007.01578 [stat.CO].
- [6] Ben Cousins y Santosh Vempala. “A practical volume algorithm”. En: *Mathematical Programming Computation* 8 (oct. de 2015). DOI: 10.1007/s12532-015-0097-z.
- [7] Ben Cousins y Santosh S. Vempala. “Gaussian Cooling and an $O^*(n^3)$ Volume Algorithm”. En: *CoRR* abs/1409.6011 (2014). arXiv: 1409.6011. URL: <http://arxiv.org/abs/1409.6011>.
- [8] Jesus De Loera, Brandon Dutra, Matthias Köppe, Stanislav Moreinis, Gregory Pinto y Jianqiu Wu. “Software for exact integration of polynomials over polyhedra”. En: *Computational Geometry* 46.3 (2013), págs. 232-252. URL: <http://www.sciencedirect.com/science/article/pii/S0925772112001113>.
- [9] Martin Dyer y Alan Frieze. “On the Complexity of Computing the Volume of a Polyhedron”. En: *SIAM J. Comput.* 17 (1988), págs. 967-974.
- [10] National Research University Higher School of Economics. *Advanced Machine Learning, Bayesian Methods for Machine Learning*. Inf. téc. URL: <https://www.coursera.org/specializations/aml>.
- [11] Ioannis Emiris y Vissarion Fisikopoulos. “Efficient Random-Walk Methods for Approximating Polytope Volume”. En: *Proceedings of the Annual Symposium on Computational Geometry* (jun. de 2014). DOI: 10.1145/2582112.2582133.
- [12] Ioannis Z. Emiris y Vissarion Fisikopoulos. “Practical Polytope Volume Approximation”. En: *ACM Trans. Math. Softw.* 44.4 (jun. de 2018). ISSN: 0098-3500. DOI: 10.1145/3194656. URL: <https://doi.org/10.1145/3194656>.
- [13] Andreas Enge. “Volume computation for polytopes: Strategies and Performances”. En: *Encyclopedia of Optimization*. Ed. por Christodoulos A. Floudas y Panos M. Pardalos. Boston, MA: Springer US, 2009, págs. 4032-4038. ISBN: 978-0-387-74759-0. DOI: 10.1007/978-0-387-74759-0_705. URL: <https://www.math.u-bordeaux.fr/~aenge/index.php?category=publications&page=chapter>.
- [14] D.S.K. Fok y Daniel Crevier. “Volume estimation by Monte Carlo methods”. En: *Journal of Statistical Computation and Simulation* 31 (abr. de 1989), págs. 223-235. DOI: 10.1080/00949658908811145.
- [15] Cunjing Ge y Feifei Ma. *A Fast and Practical Method to Estimate Volumes of Convex Polytopes*. Ed. por Jianxin Wang y Chee Yap. Springer International Publishing, 2015, págs. 52-65.
- [16] Davide Giavarina. “Understanding Bland Altman analysis.” En: *Biochemia Medica* 25.2 (2015), págs. 141-151. ISSN: 13300962. URL: <http://proxy.timbo.org.uy/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=103213206&lang=es&site=eds-live>.
- [17] Roger Guimerá y Marta Sales-Pardo. *La promesa de las redes metabólicas*. Inf. téc. 2015. URL: <https://www.sebbm.es/revista/pdf.php?id=44>.

- [18] Oliver Hädicke y Steffen Klamt. “EColiCore2: a reference network model of the central metabolism of *Escherichia coli* and relationships to its genome-scale parent model”. En: *Scientific Reports* 7 39647 (2017). URL: <https://doi.org/10.1038/srep39647>.
- [19] Martin Henk, Jürgen Richter-Gebert y Günter M. Ziegler. “Basic Properties of Convex Polytopes”. En: *Handbook of Discrete and Computational Geometry*. USA: CRC Press, Inc., 1997, págs. 243-270. ISBN: 0849385245.
- [20] Janet Iwasa y Wallace Marshall. En: *Biología celular y molecular: Conceptos y experimentos*. McGraw-Hill, 2011. Cap. 3, págs. 105-114.
- [21] Uwe Jaekel. “A Monte Carlo Method for High-Dimensional Volume Estimation and Application to Polytopes”. En: *Procedia CS* 4 (dic. de 2011), págs. 1403-1411. DOI: 10.1016/j.procs.2011.04.151.
- [22] Ingrid M. Keseler, Amanda Mackie, Alberto Santos-Zavaleta, Richard Billington, César Bonavides-Martínez, Ron Caspi, Carol Fulcher, Socorro Gama-Castro, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Luis Muñoz-Rascado, Quang Ong, Suzanne Paley, Martin Peralta-Gil, Pallavi Subhraveti, David A. Velázquez-Ramírez, Daniel Weaver, Julio Collado-Vides, Ian Paulsen y Peter D. Karp. “The EcoCyc database: reflecting new knowledge about *Escherichia coli* K-12”. En: *Nucleic Acids Research* 45.D1 (nov. de 2016), págs. D543-D550. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1003. eprint: <https://academic.oup.com/nar/article-pdf/45/D1/D543/8846544/gkw1003.pdf>. URL: <https://doi.org/10.1093/nar/gkw1003>.
- [23] Scott Kirkpatrick, C. Gelatt y M. Vecchi. “Optimization by Simulated Annealing”. En: *Science (New York, N.Y.)* 220 (jun. de 1983), págs. 671-80.
- [24] Matthias Köppe. *Software LATTE*. <https://www.math.ucdavis.edu/latte/>. Accedido 08-08-2021.
- [25] Nicolas Lartillot y Hervé Philippe. “Computing Bayes Factors Using Thermodynamic Integration”. En: *Systematic Biology* 55.2 (abr. de 2006), págs. 195-207. ISSN: 1063-5157. DOI: 10.1080/10635150500433722. eprint: <https://academic.oup.com/sysbio/article-pdf/55/2/195/26557316/10635150500433722.pdf>. URL: <https://doi.org/10.1080/10635150500433722>.
- [26] Thomas Latzko, Wolfgang Wiechert, Katharina Nöh y Uwe Jaekel. “Markov chain Monte Carlo methods to analyze the steady-state flux solution space of metabolic network models”. En: *ESM 2012 - 2012 European Simulation and Modelling Conference: Modelling and Simulation 2012* (ene. de 2012), págs. 84-90.
- [27] László Lovász. “Hit-and-run mixes fast.” En: *Math. Program.* 86, 443-461 (1999). URL: <https://doi-org.proxy.timbo.org.uy/10.1007/s101070050099>.
- [28] Iain Murray. “Advances in Markov chain Monte Carlo methods”. En: *PhD thesis* (2007). URL: http://homepages.inf.ed.ac.uk/imurray2/pub/07thesis/murray_thesis_2007.pdf.
- [29] Jeffrey Orth, R.M.T Fleming y Bernhard Palsson. “Reconstruction and Use of Microbial Metabolic Networks: the Core *Escherichia coli* Metabolic Model as an Educational Guide”. En: *EcoSal Plus* (2010). URL: <https://www.asmscience.org/content/journal/ecosalplus/10.1128/ecosalplus.10.2.1>.
- [30] Malcolm Sambridge. “A Parallel Tempering algorithm for probabilistic sampling and multimodal optimization”. En: *Geophysical Journal International* 196.1 (oct. de 2013), págs. 357-374. ISSN: 0956-540X. DOI: 10.1093/gji/ggt342. eprint: <https://academic.oup.com/gji/article-pdf/196/1/357/5963798/ggt342.pdf>. URL: <https://doi.org/10.1093/gji/ggt342>.
- [31] Pedro Sánchez Legaz. “Un inicio a la teoría de politopos”. En: (jul. de 2017). URL: https://www.um.es/documents/118351/9216932/TFG_SANCHEZ+LEGAZ_Un+inicio+a+la+teor%C3%ADa+de+politopos.pdf/a4511848-1ed2-42aa-a817-925b77c900b3.
- [32] Miklós Simonovits. “How to compute the volume in high dimension?” En: *Math. Program.* 97 (jul. de 2003), págs. 337-374. DOI: 10.1007/s10107-003-0447-x.
- [33] Robert L. Smith. “Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions”. En: *Operations Research* 32.6 (1984), págs. 1296-1308. ISSN: 0030364X, 15265463. URL: <http://www.jstor.org/stable/170949>.
- [34] Pablo Teixeira Manion. *Aplicación de métodos de Monte Carlo en el estudio de redes metabólicas*. Inf. téc. 2019. URL: <https://hdl.handle.net/20.500.12008/22168>.
- [35] Min Zhou, Fei He, Xiaoyu Song, Gangyi Chen y Ming Gu. “Estimating the Volume of Solution Space for Satisfiability Modulo Linear Real Arithmetic”. En: *Theory of Computing Systems* 56 (feb. de 2014).

Apéndice A

Algoritmos estado del arte

A continuación se incluyen descripciones de los algoritmos utilizados. Dichas descripciones no sustituyen a los artículos originales y fueron hechas en base a los mismos a modo de resumen con la única finalidad de ayudar en el proceso de estudio para comprender el funcionamiento de dichos algoritmos.

A.1. Sequence of Balls (SoB)

La descripción de este algoritmo fue tomada de [11]

Como se mencionó anteriormente, este tipo de algoritmos presenta tres etapas. Primero una etapa de redondeo que se ocupa de transformar el politopo para asegurar la convergencia. Luego viene la etapa de "sandwiching" que consiste en encontrar una esfera que contiene al politopo y otra que está incluida en el, con las cuales se van a construir los ratios a ser estimados. Finalmente se concluye aplicando el enfoque MMC.

Rounding: Aplicar una transformación lineal a P para alcanzar una posición casi isotrópica. Para el redondeo, se muestra un conjunto S de $O(n)$ puntos aleatorios en P . Luego, se aproxima el elipsoide de volumen mínimo ε que cubre S , y satisface las inclusiones $\frac{1}{(1+\varepsilon)^d}\varepsilon \subseteq \text{conv}(S) \subseteq \varepsilon$

$$\begin{aligned}\varepsilon &= \{x \in R^d | (x - c_\varepsilon)^T E (x - c_\varepsilon) \leq 1\} \\ &= \{x \in R^d | L^T (x - c_\varepsilon) \leq 1\}\end{aligned}$$

Donde $E \subseteq R^{d \times d}$ es una matriz semidefinida positiva y $L^T L$ es la descomposición de Cholesky. Sustituyendo $x = (L^T)^{-1}y + c_\varepsilon$ se mapea el elipsoide a la bola $\{y \in R^d | y^T y \leq 1\}$. Aplicando esta transformación a P , se tiene $P' = \{y \in R^d | A(L^T)^{-1} \leq b - A c_\varepsilon\}$ que es el politopo redondeado y se cumple que $\text{vol}(P) = \det(L^T)^{-1} \text{vol}(P')$. Se repite el procedimiento hasta que la relación de los ejes mínimos y máximos del elipsoide alcanza algún umbral definido por el usuario (ε).

Sandwiching: Calcular la bola B y el escalar ρ tal que $B \subseteq P \subseteq \rho B$. Primero se calcula la bola de Chebychev $B(c, r)$ de P , es decir, la bola inscrita más grande en P . Para esto, se resuelve el siguiente problema de programación lineal:

$$\begin{cases} \text{maximizar } R \\ \text{s.t.} \\ A_i x + R \|A_i\|_2 \leq b_i, \quad i = 1, \dots, m \\ R \geq 0 \end{cases}$$

donde S_i es la i -ésima fila de A , y los valores óptimos de R y $x \in R^d$ producen, respectivamente, el radio r y el centro c de la bola de Chebychev. Luego, se calcula un punto aleatorio uniforme en $B(c, r)$ y se usa como un comienzo para realizar una caminata aleatoria en P , generando N puntos aleatorios. Calculando la máxima distancia entre cada uno de los N puntos y c , se define aproximadamente una bola que contiene a P .

MMC: Finalmente, para aplicar MMC, se define la secuencia de bolas $B(c, 2^{i/d}), i = \alpha, \alpha + 1, \dots, \beta$ donde

$P_\alpha = B(c, 2^{\alpha/d}) \subseteq B(c, r)$ y P_β contiene (aproximadamente) a P . Para la generación de puntos aleatorios, utilizan RDHR y CDHR. $W = \lceil 10 + d/10 \rceil$ se define como el largo de las caminatas.

En esta implementación, a diferencia de los enfoques típicos, que generan puntos en P_i para $i = \alpha, \alpha + 1, \dots, \beta$, se procede a la inversa. Se comienza generando N puntos aleatorios dentro de P_β y luego se cuentan cuántos de ellos caen en $P_{\beta-1}$ para estimar $\text{vol}(P_\beta)/\text{vol}(P_{\beta-1})$. De esta manera es posible aprovechar los puntos que se encuentran en $P_{\beta-1}$, en la siguiente estimación $\text{vol}(P_{\beta-1})/\text{vol}(P_{\beta-2})$. Se repite el mismo procedimiento hasta calcular la última relación $\text{vol}(P_\alpha)/\text{vol}(P_{\alpha-1})$.

Algorithm 3: Volumen(P, ϵ, t_r)

```

 $N = 400\epsilon^{-2}d \log(d)$ ;
 $W = 10 + d/10$ ;
 $B(c, r) =$  Chebychev ball;
 $p$  es un punto aleatorio en  $B(c, r)$ ;
repeat
   $S = \emptyset$ ;
  for  $i = 1 \dots N$  do
     $p \leftarrow \text{Walk}(p, P, W)$ ;
    add  $p$  en  $S$ ;
    Calcular el elipsoide de volumen minimo que cubre  $S$  con  $E$ 
    Setear  $\epsilon_{min}$  y  $\epsilon_{max}$  como los ejes minimo y maximo de  $\epsilon$ 
    Calcular la descomposicion de Cholesky de  $E = L^T L$ 
    Transformar  $P$  y  $p$ 
  end
until  $\epsilon_{max}/\epsilon_{min} < t_r$ ;
 $\rho =$  distancia maxima desde  $c$  hasta cualquier punto de  $S$ ;
//MMC
 $\alpha = \log(\rho)$ ;
 $\beta = \log(\rho)$ ;
 $p_i = P \cap B(c, 2^{i/d}) \forall i = \alpha, \alpha + 1, \dots, \beta$ ;
 $\text{vol}(P_\alpha) = 2\pi^{d/2}(2^{\log(\rho)})^d/d\Gamma(d/2)$ ;
 $i = \beta$ ;
while  $i > \alpha$  do
   $P_{large} = P_i$ ;
   $i = i - 1$ ;
   $P_{small} = P_i$ ;
   $count_{prev} = \text{size}(S)$ ;
  remover de  $S$  los puntos que no esten en  $P_{small}$ ;
   $count = \text{size}(S)$ ;
  for  $j = 1, \dots, N - count_{prev}$  do
     $p = \text{Walk}(p, P_{large}, W)$ ;
    if  $p \in B(c, 2^{i/d})$  then
       $count = count + 1$ ;
      add  $p$  in  $S$ ;
    end
  end
   $vol = vol \cdot (N/count)$ 
end
Preturn  $vol/\det(L^T)$ 

```

A.2. Cooling Gaussians (CG)

La descripcion de este algoritmo fue tomada de [6].

Como se mencionó anteriormente, el cálculo del volumen es un caso particular de integración. Es decir, el volumen de P se puede expresar como

$$\text{vol}(P) = \int_P 1 \, dx$$

Si se reescribe $\text{vol}(P)$, para cualquier función $f : R^n \rightarrow R$:

$$\text{vol}(P) = \frac{\int_P f(x) \, dx}{\int_P f(x) \, dx} \int_P 1 \, dx = \int_P f(x) \, dx \frac{\int_P 1 \, dx}{\int_P f(x) \, dx}$$

Se reduce la tarea del cálculo del volumen a resolver dos problemas: calcular la razón de dos integrales e integrar la función f sobre P . Esta representación se puede extender a cualquier secuencia de funciones $\{f_0, \dots, f_{m-1}\}$ donde cada $f_i : R^n \rightarrow R$.

$$\text{vol}(P) = \int_P f_0(x) dx \frac{\int_P f_1(x) dx}{\int_P f_0(x) dx} \frac{\int_P f_2(x) dx}{\int_P f_1(x) dx} \cdots \frac{\int_P 1 dx}{\int_P f_{m-1}(x) dx}$$

Elijiendo las funciones f_i de modo que cada término de la ecuación anterior se pueda calcular de manera eficiente, es que se llega al algoritmo CG implementado en la librería VolEsti. Se define f_0 como un gaussiano de baja varianza centrado dentro de P , esto hace que el peso del gaussiano quede concentrado alrededor de un solo punto dentro de P . Entonces, si el gaussiano es lo suficientemente "nítido", se cumple que $\int_P f_0(x) dx \approx \int_{R^n} f_0(x) dx$. Como la integración de un gaussiano sobre R^n es sencilla, lo que se logra es poder aproximar el término $\int_P f_0(x) dx$ de manera directa.

En la estimación de cada relación integral $\int_P \frac{f_i(x) dx}{\int_P f_{i-1}(x) dx}$ es donde entra en juego el muestreo. Si se puede obtener una muestra aleatoria de una distribución proporcional a f_{i-1} restringida a P , se denota la medida de esta muestra aleatoria como μ_{i-1} . Entonces, para una muestra aleatoria X extraída de μ_{i-1} , se define una variable aleatoria $Y = \frac{f_i(X)}{f_{i-1}(X)}$, el valor esperado de esta variable aleatoria, es igual a la cantidad deseada que se quiere estimar:

$$E(Y) = \int_P \frac{f_i(x)}{f_{i-1}(x)} d\mu_{i-1}(x) = \int_P \frac{f_i(x)}{f_{i-1}(x)} \frac{f_{i-1}(x)}{\int_P f_{i-1}(x)} dx = \frac{\int_P f_i(x) dx}{\int_P f_{i-1}(x) dx}$$

Y esta estimación se consigue a partir de las k muestras $\{X_1, \dots, X_k\}$. Entonces, se puede estimar la razón integral como

$$\frac{1}{k} \sum_{j=1}^k \frac{f_i(X_j)}{f_{i-1}(X_j)}$$

Para que la estimación sea eficiente, lo importante es que el número de muestras k que se necesitan no sea demasiado grande. Si se pasa inmediatamente de f_0 a la distribución uniforme, se requeriría un número exponencial de muestras k . Por esto es que se introduce la idea de "Annealing schedule" que consiste en encontrar una configuración adecuada de funciones f_i que permitan ir "enfriando" a f_0 hasta llegar a la distribución uniforme, mientras se controla la varianza de Y , lo que da como resultado un pequeño número de fases (es decir, relaciones integrales) y no demasiadas muestras por fase. En esta implementación eso se logra seleccionando a cada f_i como un gaussiano, y aumentando lentamente la varianza hasta que el gaussiano sea esencialmente la distribución uniforme (es decir, el volumen).

Al igual que en el caso del algoritmo SoB, se tienen tres etapas. Primero la etapa de redondeo. Luego lo que vendría a ser la etapa de "sandwiching" que en este caso es la etapa del "Programa de enfriamiento" (Annealing Schedule) y finalmente la etapa donde se aplica la idea detrás de MMC.

Rounding: Para obtener las muestras aleatorias X extraída de μ_{i-1} , se utiliza Hit-and-Run. Se sabe que

si se cumple que $rB_n \subseteq P \subseteq RB_n$ entonces se requieren $O * (n^2 R^2 / r^2)$ pasos de hit-and-run para que el algoritmo converja a la distribución objetivo. Por este motivo es que la relación R/r afecta la performance del algoritmo, si R/r es muy grande se necesitan muchos pasos de Hit-and-Run para llegar a la distribución objetivo. Intuitivamente, esto se debe a que lleva mucho tiempo moverse de un extremo al otro dentro de un politopo alargado, mientras que es comparativamente más fácil moverse en un politopo redondeado. Por lo tanto, al igual que en SoB es necesario aplicar un paso de redondeo al politopo antes de comenzar con la estimación del

volumen. El algoritmo para el redondeo es el siguiente:

Algorithm 4: Round(P, t)

Entradas: P cuerpo convexo, t cantidad pasos (Nota: se asume que $B_n \subseteq P \subseteq RB_n$) $x_0 = 0$;
 $T = I$; $tries = 0$;
repeat
 $tries = tries + 1$;
 for $i=1, \dots, t$ **do**
 $x_i = HitAndRun(TP, f = 1, x_{i-1})$;
 end
 $(U, S, V^T) = SVD(x_1, \dots, x_t)$;
 $T = SV^{-1}T$;
until $max(S) \leq 2$ or $tries > \log(R)$;
if $max(S) \leq 2$ **then**
 return T ;
else
 return Round($P, 2t$);
end

El objetivo es igual que en SoB, llevar al politopo a una posición casi isotrópica, pero en este caso se maneja la definición de función de densidad isotrópica y lo que se hace es llevar un conjunto de puntos, muestreados a partir de esta distribución a una posición casi isotrópica. Se define que una función de densidad es isotrópica si su centroide es 0 y su matriz de covarianza es la identidad. Es decir, para una variable aleatoria X extraída de f , se cumple que $E(X) = 0$ y $E(XX^T) = I$. Lo que es equivalente a decir que para todo vector unitario $v \in R^n$ se cumple que $\int_{R^n} (v^T x)^2 f(x) dx = 1$. A partir de esta definición, se considera la noción de aproximadamente isotrópico, y se dice que f es C -isótropica si $\frac{1}{C} \leq \int_{R^n} (v^T x)^2 f(x) dx \leq C$.

Lo que se hace es redondear el cuerpo con respecto a la distribución uniforme. El objetivo es encontrar una transformación lineal T tal que TP esté en posición 2-isotrópica. Esto se hace obteniendo una secuencia de puntos $\{X_1, X_2, \dots, X_r\}$ mediante la aplicación de hit-and-run (con la distribución uniforme como distribución objetivo) sobre P . Luego se calcula la transformación que pone este conjunto de puntos en una posición aproximadamente isotrópica. Si se realiza la cantidad suficiente de pasos hit-and-run, entonces al aplicar la transformación lineal encontrada a el politopo $P' = TP$ lo deja en una posición aproximadamente isotrópica. Para $n \leq 100$ los autores plantean que en sus experimentos observaron que con $8n^3$ muestras de hit-and-run es suficiente llevar a P a una posición 2-isotrópica. Pero en el caso de que llegado este punto, se note que el cuerpo no está lo suficientemente redondeado, se vuelve a repetir el proceso con el doble de muestras hasta llegar a una convergencia.

Para calcular la transformación lineal que pone a los puntos $\{X_1, X_2, \dots, X_r\}$ muestreados en posición isotrópica, primero se hace la descomposición en valores singulares de estos puntos. Es decir, para la matriz de puntos M , se encuentran matrices tales que $M = USV^T$ con $M \in R^{m \times n}$ tal que S es una matriz diagonal que contiene los n valores singulares s_i , $U \in R^{m \times n}$ y $V \in R^{n \times n}$ cuyas columnas forman una base ortonormal de R^m y R^n respectivamente. Los valores singulares de una matriz son las longitudes de los semiejes de la hiper-elipse en que se convierte la esfera de unidad si le aplicamos dicha matriz como una transformación lineal. Intuitivamente podemos observar que cuanto menos se transforme la esfera en una elipse, más redondeado va a estar el politopo. Por otro lado, la parte de la descomposición USV^T que produce la deformación en una hiper-elipse cuando se aplica a la esfera de unidad es SV^T entonces si denotamos a la esfera de unidad como B se puede decir que lo que se busca es que:

$$SV^T B \approx B = S^{-1}SV^T B \approx S^{-1}B = V^T B \approx S^{-1}B = VV^T B \approx VS^{-1}B = B \approx VS^{-1}B$$

Los valores singulares s_i da la coordenada de P en la dirección $u_i v_i^T$. Entonces en otras palabras, lo que se está haciendo es escalar al politopo a lo largo de las direcciones determinadas por $u_i v_i^T$. (Los valores singulares más pequeños se ignoran por estabilidad numérica) Otro punto importante a destacar es que los autores suponen que P está contenido en una bola de radio R , porque los autores observaron que si se toman suficientes puntos de muestra, cada redondeo debería reducir el valor singular máximo en un factor constante. Si no hay convergencia dentro de $\log R$ iteraciones, se entiende que el número de pasos t no fue suficiente para estimar con precisión los valores singulares, por lo que se reinicia el procedimiento con $2t$ pasos.

Programa de enfriamiento Como mencionamos más arriba, en esta implementación para establecer la secuencia apropiada de funciones $\{f_0, f_1, \dots, f_m\}$ (Annealing schedule), se selecciona a cada f_i como un gaussiano,

y se va aumentando lentamente la varianza hasta que el gaussiano sea la distribución uniforme. Para esto, primero se determina f_0 para que $\int_P f_0 dx$ sea fácil de calcular. Y a partir de f_0 se utiliza enfoque recursivo que calcula f_i a partir de f_{i-1} . Entonces si se toma $f_0 = e^{-a\|x\|^2}$, se puede encontrar un a_0 que cumpla que, dado un cuerpo convexo K y un parámetro de error $\varepsilon > 0$:

$$\int_K e^{-a_0\|x\|^2} dx \geq (1 - \varepsilon) \int_{\mathbb{R}^n} e^{-a_0\|x\|^2} dx$$

Se asume que $K = P \cap E$, P es el politopo para el cual se quiere calcular el volumen y E es un elipsoide.

Para encontrar el valor a_0 que permita que se cumpla esta desigualdad, lo que se hace es limitar la probabilidad en función de a denotada por $p(a)$, de que $X \notin K$ usando los lemas A.2.1 y A.2.2 y la descripción del cuerpo K .

Lemma A.2.1. Supongamos que X se extrae de un gaussiano esférico en \mathbb{R}^n con media μ y varianza σ^2 en cualquier dirección. Entonces, para cualquier $t > 1$,

$$P(\|X - \mu\| - \sigma^2 n > t\sigma^2 \sqrt{n}) \leq e^{-t^2/8}$$

Lemma A.2.2. Supongamos que X se extrae de un Gaussiano unidimensional con varianza $\sigma^2 = 1/(2a)$. Entonces, se cumple que,

$$P(X > t) \leq \frac{e^{-at^2}}{2t\sqrt{a\pi}}$$

Primero se calcula la distancia mínima desde $0 \in K$ a cada hiperplano que describe a P y al límite de E . Luego se aplica el lema uno al elipsoide y el lema dos al politopo y la unión ligada a estas probabilidades para obtener un límite inferior para la fracción del gaussiano que se encuentra dentro del cuerpo convexo K . Es decir, si $d(0, H)$ es la distancia mínima desde 0 a la superficie del elipsoide,

$$\begin{aligned} P(x \notin K) &\leq P(x \notin P) + P(x \notin E) \leq \sum_H P(x \text{ no cumple con } H) + P(\|x\| > d) \\ &\leq \sum_H P(\|x_H\| > d(0, H)) + e^{-d^2/8} \\ &\leq \sum_H \frac{e^{-a \cdot d(0, H)^2}}{2d(0, H)\sqrt{a\pi}} + e^{-d^2/8} =: p(a) \end{aligned}$$

donde x_H es la proyección de x sobre la normal de H . Finalmente, mediante una búsqueda binaria se obtiene el valor a_0 que cumple que $p(a_0) = \varepsilon$

A continuación presentamos el pseudocódigo de este algoritmo:

Algorithm 5: FirstGaussian. Devuelve a_0

```

Entradas:  $P$  y  $\varepsilon$ ;
 $dists = d(0, H)$ ;
 $lower = 0$ ;
 $upper = 0$ ;
 $mid = 0$ ;
 $fin = false$ ;
 $tol = 0,001$ ;
se busca el limite superior para  $p(a) = P(x \notin K)$ ;
while not  $fin$  do
     $sum = \sum_{d \in dists} \frac{e^{-upper \cdot d^2}}{2d\sqrt{\pi \cdot upper}}$ ;
    if  $sum > \varepsilon$  then
         $upper = upper \cdot 10$ ;
    else
         $fin = true$ ;
    end
end
Busqueda binaria de  $a_0$ ;
while  $upper - lower > tol$  do
     $mid = (upper + lower)/2$ ;
     $sum = \sum_{d \in dists} \frac{e^{-mid \cdot d^2}}{2d\sqrt{\pi \cdot mid}}$ ;
    if  $sum < \varepsilon$  then
         $upper = mid$ ;
    else
         $lower = mid$ ;
    end
end
return  $a_0 = (upper + lower)/2$ ;

```

El siguiente paso es construir la secuencia de funciones f_i que convergen a la distribución uniforme sobre K de manera que sea posible estimar los ratios:

$$R_i = \frac{\int_K f_i(x) dx}{\int_K f_{i-1}(x) dx} = \frac{\int_K e^{-a_i \|x\|^2} dx}{\int_K e^{-a_{i-1} \|x\|^2} dx}$$

Esto se hace a partir de la función $f_0 = e^{-a_0 \|x\|^2}$ y teniendo en cuenta los resultados presentados en el siguiente lema:

Lemma A.2.3. Sea X un punto aleatorio en K con densidad proporcional a $e^{-a_i \|x\|^2}$, $a_{i+1} = a_i(1-1/n)$, $n \geq 4$, y

$$Y = e^{(a_{i+1} - a_i) \|X\|^2}$$

entonces:

$$\frac{Var(Y)}{E(Y)^2} \leq \left(\frac{a_i^2}{a_{i+1}(2a_i - a_{i+1})} \right)^{n+1} = \left(1 + \frac{1}{n^2 - 2n} \right)^n + 1 < 1$$

$$Y = \frac{f_i(X)}{f_{i-1}(X)} = \frac{e^{-a_i \|x\|^2}}{e^{-a_{i-1} \|x\|^2}} = e^{(a_{i-1} - a_i) \|X\|^2}$$

$$E(Y) = \frac{\int_K f_i(x) dx}{\int_K f_{i-1}(x) dx}$$

La idea es seleccionar los $\{f_i\}$ que hagan que la convergencia a la distribución uniforme sea lo más rápida posible, manteniendo la varianza de Y controlada (con el fin de mantener el cociente entre integrales, para conseguir llegar al volumen en pocas fases). Esto se hace tratando de maximizar k , donde $a_{i+1} = a_i(1-1/n)^k$.

Por el lema A.2.3, se comienza con $k = 1$ y a partir de un número pequeño muestreado utilizando HR (con distribución objetivo proporcional a $f_{i-1} \cap K$) y observando su varianza se busca el k máximo que cumpla que

$\frac{Var(Y)}{E(Y)^2} \leq 1$. En la práctica, los autores de la implementación de este algoritmo observa que aproximadamente $O(n^2)$ pasos de HR son suficientes para estimar $\frac{Var(Y)}{E(Y)^2}$.

A continuación presentamos el pseudocódigo de este algoritmo:

Algorithm 6: NexGaussian. Devuelve $a_{i+1} = a_i(1 - 1/n)^k$

```

Entradas:  $x_0 \in P$  ;
y  $a_i$  valor anterior;
N cantidad de muestras para HR;
 $k = 1$ ;
 $X = HitAndRun(x_0, P)$  Muestrear N puntos usando HR;
 $fin = false$ ;
while not fin do
     $a_{i+1} = a_i(1 - 1/n)^k$ ;
     $Y = \frac{f_{i+1}(X)}{f_i(X)} = \frac{e^{-a_{i+1}\|x\|^2}}{e^{-a_i\|x\|^2}} = e^{(a_i - a_{i+1})\|x\|^2}$ ;
    calcular  $Var(Y)$  y  $E(Y)$ ;
    if  $Var(Y)/E(Y)^2 \geq 1$  then
        if  $k \neq 1,0$  then
             $k = k/2$ 
        end
         $fin = true$ ;
    else
         $k = 2k$ ;
    end
end

```

Una vez que se verifica que f_i está lo suficientemente cerca de la distribución uniforme, se detiene la búsqueda de los a_i .

Algorithm 7: GetAnnealingSchedule. Devuelve $\{a_0, \dots, a_m\}$

```

 $a_0 = FirstGaussian(P, \varepsilon)$ ;
 $fin = false$ ;
 $a_{i-1} = a_0$ ;
while not fin do
     $a_i = NextGaussian(P, a_{i-1})$  Verificar convergencia a la distribución normal;
     $samples =$  muestras tomadas de la distribución  $f_i$ ;
     $curr_{its} = \sum_{p \text{ in } samples} 1$ ;
     $curr_{fn} = \sum_{p \text{ in } samples} e^{-a_i\|p\|^2}$ ;
    if  $a_i > 0$  and  $curr_{fn}/curr_{its} > 1$  then
         $a_{i-1} = a_i$ ;
    else
         $fin = true$ ;
    end
end
return  $\{a_0, \dots, a_m\}$ 

```

Finalmente, tenemos todo como para mostrar el pseudocódigo del algoritmo del cálculo de volumen. Incluyendo la etapa de MMC, en donde se estiman los ratios R_i de las integrales.

Algorithm 8: Volumen(P, ε)

```

 $T = \text{Round}(P, 8n^3);$ 
 $P' = T \cdot P;$ 
 $\{a_0, \dots, a_m\} = \text{GetAnnealingSchedule}(P, \varepsilon);$ 
 $x$  es un punto aleatorio de  $f_0 \cap P', \varepsilon' = \varepsilon/\sqrt{m};$ 
for  $i=1, \dots, m$  do
   $k = 0;$ 
   $x_0 = x;$ 
   $W = 4n^2 + 500;$ 
   $fin = \text{false};$ 
  while not  $fin$  do
     $k = k + 1;$ 
     $x_k = \text{HitAndRun}(P', \text{distribucion objetivo} : f_{i-1}, x_{k-1});$ 
     $r_k = \frac{1}{k} \sum_{j=1}^k \frac{f_i(x_j)}{f_{i-1}(x_j)};$ 
     $W_{max} = \max\{r_{k-W+1}, \dots, r_k\};$ 
     $W_{min} = \min\{r_{k-W+1}, \dots, r_k\};$ 
    if  $(W_{max} - W_{min} \leq \varepsilon'/2 \cdot W_{max})$  then
       $fin = \text{true};$ 
    end
  end
   $R_i = r_k;$ 
   $x = x_k;$ 
end
return volumen =  $|T| \cdot (\pi/a_0)^{n/2} \cdot R_1 \cdots R_m$ 

```

A.3. Cooling Bodies (CB)

La descripción de este algoritmo fue tomada de [4].

El algoritmo se basa en MMC y utiliza Hit and run para generar muestras con la distribución uniforme como objetivo. Se reescribe el volumen de un politopo P mediante la siguiente ecuación:

$$\text{vol}(P) = \frac{\frac{\text{vol}(P_m)}{\text{vol}(C_m)}}{\frac{\text{vol}(P_1)}{\text{vol}(P_0)} \frac{\text{vol}(P_2)}{\text{vol}(P_1)} \cdots \frac{\text{vol}(P_m)}{\text{vol}(P_{m-1})}} \text{vol}(C_m) \quad \text{con } P_0 = P, P - i = C_i \cap P \quad (\text{A.1})$$

A cada ratio de esta ecuación se los denomina r_i :

$$r_i = \frac{\text{vol}(P_{i+1})}{\text{vol}(P_i)}, \quad i = 0, \dots, m-1 \quad \text{y} \quad r_m = \frac{\text{vol}(P_m)}{\text{vol}(C_m)}$$

Primero se construye una secuencia de cuerpos convexos $C_1 \supseteq \cdots \supseteq C_m$ que se intersectan con el politopo P usando “enfriamiento simulado”. Luego, se estima cada relación en el producto telescópico de la Ecuación A.1 usando HR y un criterio empírico de convergencia. En estimaciones prácticas C_m tiene que ser un cuerpo convexo cuyo volumen se obtiene mucho más rápido que $\text{vol}(P)$ y fácil de muestrear. Una opción típica para los C_i es una secuencia de bolas concéntricas, pero se pueden usar otros cuerpos convexos.

Programa de enfriamiento Dado un politopo convexo P , un parámetro de error ε y r, δ, α tales que

$0 < r + \delta < 1$, el programa de enfriamiento genera la secuencia de cuerpos convexos $C_1 \supseteq \cdots \supseteq C_m$ que definen los $P_i = C_i \cap P$, $i = 1, \dots, m$ y $P_0 = P$. El objetivo principal es restringir cada cociente r_i en el intervalo $[r, r + \delta]$ con alta probabilidad. Para esto, se construyen dos pruebas de hipótesis que permiten afirmar con alta probabilidad que estas restricciones sobre r_i se cumplen.

Una prueba de hipótesis es una regla que especifica si se puede aceptar o rechazar una afirmación acerca de una población dependiendo de la evidencia proporcionada por una muestra de datos. En este tipo de pruebas se examinan dos hipótesis opuestas sobre una población: la hipótesis nula H_0 y la hipótesis alternativa H_1 . La hipótesis nula es la afirmación que se está comprobando. La hipótesis alternativa es la afirmación que se desea ser capaz de concluir que es verdadera basándose en la evidencia proporcionada por los datos de la muestra. A partir de una muestra aleatoria se construye un estadístico, y según el valor que tome este estadístico de prueba

se aceptará o se rechazará la hipótesis. El estadístico es el valor que se quiere extrapolar a la población de la muestra.

Como se está tomando una decisión en base a el valor de un estadístico se puede cometer dos tipos de errores: rechazar H_0 cuando ésta es verdadera, o aceptar H_0 cuando ésta es falsa. El primero se conoce como error de tipo I, y el segundo como error de tipo II. A $\alpha = P(\text{rechazar } H_0 \mid H_0 = \text{true})$ se lo conoce como nivel de significancia del test. A menor valor de α , menor probabilidad de rechazar la hipótesis nula. Si el estadístico es menor que el valor de α seleccionado, existen evidencias suficientes para rechazar H_0 en favor de H_1 . La probabilidad de cometer un error del tipo dos es igual a β (figura A.1). En Particular, las pruebas que son utilizadas

| | H_0 Verdadera | H_0 Falsa |
|---------------------|---|--|
| Rechazamos H_0 | Error Tipo I P(error Tipo I) = α | Decisión Correcta |
| No Rechazamos H_0 | Decisión Correcta | Error Tipo II P(error Tipo II) = β |

Figura A.1: Probabilidades de error en pruebas de hipótesis.

en el desarrollo de este algoritmo, se aplican si X_1, \dots, X_ν son variables aleatorias independientes distribuidas normalmente, con media μ , varianza σ y media muestral $\bar{X} = \frac{(X_1 + \dots + X_\nu)}{\nu}$ y utilizan el estadístico

$$T = \frac{\bar{X}_\nu - \mu}{S_\nu / \sqrt{\nu}}$$

con

$$S^2(x) = \frac{1}{\nu - 1} \sum_{i=1}^n (x_i - \bar{x})^2$$

que sigue la distribución t - Student¹ con $\nu - 1$ grados de libertad. La distribución T-student es una distribución de probabilidad que surge del problema de estimar la media de una población normalmente distribuida cuando el tamaño de la muestra es pequeño. Esta distribución es muy similar a la distribución normal. Tiene como parámetros la media, la varianza y además incorpora a través de los grados de libertad una modificación que permite flexibilizar las colas en función del tamaño que tenga la muestra. A medida que se reduce el tamaño muestral, la probabilidad acumulada en las colas aumenta, siendo así menos estricta de lo que cabría esperar en una distribución normal. Una distribución T-student con 30 o más grados de libertad es prácticamente igual a una distribución normal.

Entonces, dadas ν observaciones de una variable aleatoria $X \sim N(\mu, \sigma^2)$ con varianza desconocida σ^2 , las pruebas t-test (de una cola) verifican la hipótesis nula de que la media de una población supera un valor específico μ_0 usando el estadístico $t = \frac{\hat{x} - \mu_0}{s / \sqrt{\nu}} \sim t_{\nu-1}$, donde \hat{x} es la media muestral, s la desviación estándar muestral y $t_{\nu-1}$ es la distribución t-student con $\nu - 1$ grados de libertad.

Dado el valor de significancia $\alpha > 0$ se prueba la hipótesis nula para el valor medio de la población, $H_0 : \mu \leq \mu_0$ contra $H_1 : \mu > \mu_0$. Si se cumple que $t \geq t_{\nu-1, \alpha} \Rightarrow \bar{x} \geq \mu_0 + t_{\nu-1, \alpha} s / \sqrt{\nu}$, entonces se rechaza H_0 . Lo que implica que $\alpha = P(\text{rechazar } H_0 \mid H_0 = \text{true})$.

El algoritmo para encontrar el cronograma de enfriamiento utiliza las siguientes dos pruebas estadísticas:

| | |
|--|--|
| $testL(P_1, P_2, r, \delta, \alpha, \nu, N)$ | $testR(P_1, P_2, r, \delta, \alpha, \nu, N)$ |
| $H_0 : vol(P_2)/vol(P_1) \geq r + \delta$ | $H_0 : vol(P_2)/vol(P_1) \leq r$ |
| $H_1 : vol(P_2)/vol(P_1) \leq r + \delta$ | $H_1 : vol(P_2)/vol(P_1) \geq r$ |
| <i>if H_0 is rejected return true</i> | <i>if H_0 is rejected return true</i> |

¹https://es.wikipedia.org/wiki/Distribución_t_de_Student

Entonces, si se rechazan las hipótesis nulas en ambas pruebas se cumple con el objetivo de restringir cada razón r_i en el intervalo $[r, r + \delta]$ con alta probabilidad.

Finalmente, si se muestrean N puntos uniformes en un cuerpo P_i , entonces la variable aleatoria X , que cuenta los puntos que caen en P_{i+1} , sigue una distribución binomial, $X \sim b(N, r_i)$ y $Y = X/N \sim N(r_i, r_i(1 - r_i)/N)$ es un gaussiano. (Esto pasa solo si N es lo suficientemente grande, entonces en el algoritmo se adopta la regla $Nr_i(1 - r_i) > 10$ para garantizar que esto pase). Entonces, cada proporción muestral que cuenta los éxitos en P_{i+1} sobre N es un estimador insesgado para la media de Y , que es r_i . Si se muestrean νN puntos de P_i y se divide la muestra en ν sublistas de longitud N , las relaciones ν correspondientes son valores experimentales que siguen la distribución $N(r_i, r_i(1 - r_i)/N)$ y se pueden usar para verificar ambas hipótesis nulas en testL y testR.

Algorithm 9: Aplicar pruebas testL y testR

Entradas:

P_1, P_2 cuerpos convexos;

r, δ parametros de enfriamiento;

$\alpha > 0$ nivel de significancia;

$\nu, N > 0$;

1. Tomar νN muestras uniformemente distribuidas de P_1 utilizando *HR*;
 2. Dividir las νN en ν listas S_1, \dots, S_ν de largo N ;
 3. Calcular $\hat{r}_i = |\{q \in P_2 : q \in S_i\}|/N$, $i = 1, \dots, \nu$
 4. Calcular la media $\hat{\mu}$ y la desviacion estandar s de los ν ratios.
 5. **if** $\hat{\mu} \geq r + t_{\nu-1, \alpha} \frac{s}{\sqrt{\nu}}$ **then**
| pasa testR
else
| falla testR
end
 6. **if** $\hat{\mu} \leq r + \delta - t_{\nu-1, \alpha} \frac{s}{\sqrt{\nu}}$ **then**
| pasa testL
else
| falla testL
end
-

Cada cuerpo convexo $C_i \in \{C_1 \supseteq \dots \supseteq C_m\}$ es un cuerpo convexo C multiplicado por un escalar. En el caso de los *H-politopos*, C es la esfera unitaria, pero podría ser cualquier otro cuerpo convexo.

Para encontrar la secuencia el algoritmo recibe un cuerpo convexo C (la esfera unitaria en el caso que estamos estudiando) y un intervalo $[q_{min}, q_{max}]$. Este intervalo cumple que la probabilidad de que el $testR(q_{min}C, q_{min}C \cap P)$ falle es igual a uno (esto pasa cuando $q_{min}C \subseteq P$), y la probabilidad de que el $testL(q_{max}C, q_{max}C \cap P)$ falle esta arbitrariamente cercana a uno. Cuando C es la esfera unitaria, para obtener el intervalo inicial, se establece

$q_{min} = 0$ y se muestrean νN puntos para establecer q_{max} de modo que todos los puntos pertenezcan a $q_{max}C$.

Algorithm 10: AnnealingSchedule ($P, C, r, \delta, \alpha, n, N, q_{min}, q_{max}$)

Entradas:

$q_1 = q_{min}, q_2 = q_{max}, q = (q_1 + q_2)/2;$

$stop = false;$

while not stop **do**

$C' = qC;$

if testR($C', C' \cap P$) and testL($C', C' \cap P$) **then**

$stop = true$

end

if testR($C', C' \cap P$) and not testL($C', C' \cap P$) **then**

$q_1 = q, q = (q_1 + q_2)/2$

end

if not testR($C', C' \cap P$) and testL($C', C' \cap P$) **then**

$q_2 = q, q = (q_1 + q_2)/2$

end

if not testR($C', C' \cap P$) and not testL($C', C' \cap P$) **then**

 Muestrear νN puntos con HR en C'

end

end

$P_0 = P, i = 0, q_{min} = q;$

$stop = false;$

while not stop **do**

$q_1 = q_{min}, q_2 = q_{max}, q = (q_1 + q_2)/2;$

if testR($C', C' \cap P$) **then**

$m = i + 1; P_m = C' \cap P; stop = true;$

end

$stop2 = false$

while not stop2 **do**

$P' = qC \cap P$

if testR($P_i, C' \cap P$) and testL($C', C' \cap P$) **then**

$P_{i+1} = P'; i = i + 1; stop2 = true$

end

if testR(P_i, P') and not testL(P_i, P') **then**

$q_2 = q, q = (q_1 + q_2)/2$

end

if not testR(P_i, P') and testL(P_i, P') **then**

$q_1 = q, q = (q_1 + q_2)/2$

end

if not testR($C', C' \cap P$) and not testL($C', C' \cap P$) **then**

 Muestrear νN puntos con HR en P'

end

end

$q_{max} = q$

end

$return \{P_0, \dots, P_m, C'\}$

Mediante una búsqueda binaria y utilizando los testR y testL el algoritmo encuentra el cuerpo $C' = C_m$ tal que tiene una alta probabilidad de que $r_m = vol(C_m \cap P)/vol(C_m) \in [r, r + \delta]$. Una vez que se tiene definido $C' = C_m$, se establece $P_0 = P$ y mediante otra búsqueda binaria se va encontrando la secuencia completa.

En el paso i , el algoritmo determina P_{i+1} encontrando el número q que hace que la relación de volumen se encuentre acotada con alta probabilidad, $r_i \in [r, r + \delta]$. Se toman νN muestras en P_i y se realizan búsquedas binarias para encontrar q en un intervalo actualizado $[q_{min}, q_{max}]$ de modo que tanto $testL(P_i, qC \cap P)$ como $testR(P_i, qC \cap P)$ tengan éxito. Luego, se establece $P_{i+1} = qC \cap P$. Para actualizar el intervalo, el algoritmo usa el valor q de C calculado en el paso de inicialización como q_{min} y el valor q de P_i calculado en el paso anterior como q_{max} . El intervalo actualizado implica que $vol(P_{i+1})$ tiene que estar entre $vol(P_i)$ y $vol(C \cap P)$. La búsqueda de cuerpos P_i se detiene cuando $vol(C \cap P)/vol(P_i)$ es lo suficientemente grande de acuerdo con el testR. El cuerpo $C \cap P$ es el que tiene volumen mínimo en la secuencia.

Cabe destacar que en las pruebas testR y testL del algoritmo se pueden tener errores de tipo I o II y, esto

provocaría que se hagan búsquedas binarias en intervalos que no contienen valores correspondientes en $[r, r + \delta]$. Por lo tanto, existe la probabilidad de que el algoritmo no termine. Sin embargo, los autores del algoritmo proporcionan una prueba de que esta probabilidad está limitada por una constante cuando el algoritmo realiza al menos tantos pasos como el número mínimo requerido para que termine. Dicha constante depende de las probabilidades de cometer los errores del tipo I y II en los tests.

Finalmente si se cumplen las siguientes tres condiciones: el politopo P es tal que se cumple que $rB_n \subseteq P \subseteq RB_n$, se toman muestras de puntos uniformes en cada paso del algoritmo y este termina con éxito, se demuestra que la cantidad de fases m que determina el algoritmo está limitada. En particular si se usan esferas en MMC entonces $m = O(\lg(\text{vol}(P)/\text{vol}(rB_n)))$

Después de tener la secuencia de P_i viene la parte de estimar cada ratio para poder estimar el volumen. En este algoritmo se establece un criterio de convergencia empírico y se aproxima este ratio utilizando una cantidad de puntos variable que depende de dicho criterio.

Primero, se acota el error en la estimación de cada ratio para usarlo en la definición del criterio de detención. Para cada r_i , el error esta limitado por ϵ_i de modo que:

$$\sum_{i=0}^m \epsilon_i^2 = \epsilon^2$$

Si se asume un muestreo uniforme, entonces el número de puntos en P_{i+1} sigue la distribución binomial $b(n, r_i)$, donde n es el número de puntos que se generaron en P_i . Entonces, un intervalo de confianza de r_i viene dado por

$$\hat{r} \pm z_{\alpha/2} \sqrt{\frac{\hat{r}(1-\hat{r})}{n}}$$

Dónde \hat{r} es la proporción de la cantidad de puntos en P_{i+1} sobre n y $z_{\alpha/2}$ es el cuantil $1 - \alpha/2$ de la distribución Gaussiana. Mientras n aumenta, el intervalo se vuelve más estrecho alrededor de \hat{r} , de modo que una buena opción es detenerse cuando $z_{\alpha/2} \sqrt{\frac{\hat{r}(1-\hat{r})}{n}} / (\hat{r} - z_{\alpha/2} \sqrt{\frac{\hat{r}(1-\hat{r})}{n}}) \leq \epsilon_i$ y muestrear $O(1/\epsilon_i^2)$ puntos de P_i . Con este criterio de convergencia, la ecuación A.1 estimará el volumen de P con un error acotado por ϵ con probabilidad $(1 - \alpha)^{m+1}$

Sin embargo, en la práctica no es posible utilizar este criterio porque se generan muestras aproximadamente uniformes y para caminatas aleatorias cortas el número de puntos que caen en P_{i+1} no sigue la distribución binomial y el criterio suele resultar en falsos positivos. Debido a esto, los autores del algoritmo desarrollaron un criterio empírico que utiliza la desviación estándar de una cantidad k de estimadores consecutivos de r_i . Se almacenan los últimos k r_i en una cola llamada ventana deslizante denotada por W . Se actualiza W cada vez que se genera un nuevo punto de muestra insertando el nuevo valor de r_i y sacando el valor más antiguo. A partir de la desviación estándar de estos valores y del valor del estimador \hat{r} se establece el criterio de convergencia. El tamaño k de la ventana deslizante se determina experimentalmente

Algorithm 11: EstimateRatio(P_1, P_2, e, m, k)

```


$p = 1 - \sqrt[m+1]{3/4}$ , convergence = false,  $j = 0$ , count_in = 0,  $W$  ventana de largo  $k$

while not convergence do
   $j = j + 1$ 
  Obtener un punto uniforme  $q_j$  de  $P_1$ 
  if  $q_j \in P_2$  then
    | count_in = count_in + 1
  end
   $\hat{r} = \frac{\text{count\_in}}{j}$ 
  if  $W$  esta completo then
    | remover  $W_1$ 
  end
  if  $j > k$  then
    |  $s = \text{std}(W)$ ,  $a = \hat{r} - z_{p/2}s$ ,  $b = \hat{r} + z_{p/2}s$ 
    | if  $(b - 1)/a \geq e/2$  then
      | | convergence = true
    | end
  end
end
return  $\hat{r}$ 
```

A continuación se muestra el algoritmo para el cálculo del volumen:

Algorithm 12: Volume($P, \epsilon, r, \delta, \alpha, \nu, N, k$)

Construir el cuerpo convexo $C \subseteq \mathbb{R}^n$ s.t. $C \cap P \neq \emptyset$ y definir el intervalo $[q_{min}, q_{max}]$

$\{P_0, \dots, P_m, C_m\} = \text{AnnealingSchedule}(P, C, r, \delta, \alpha, \nu, N, q_{min}, q_{max})$

Setear $\epsilon_i, i = 0, \dots, m$ s.t. $\sum_{i=0}^m \epsilon_i^2 = \epsilon^2$

for $i = 0, \dots, m$ **do**

if $i < m$ **then**

$r_i = \text{EstimateRatio}(P_i, P_{i+1}, \epsilon_i, m, k)$

else

$r_i = \text{EstimateRatio}(C_m, P_m, \epsilon_m, m, k)$

end

end

return $\text{vol}(C_m)/r_0/\dots/r_{m-1}r_m$

El comportamiento de este algoritmo está parametrizado por el error de aproximación ϵ , parámetros de enfriamiento $0 < r + \delta < 1$, $r, \delta > 0$, que se utilizan para generar la secuencia de cuerpos convexos, nivel de significación $\alpha > 0$ de las pruebas estadísticas, los grados de libertad para el t-student usados en las t-tests ν , y el parámetro N que controla el número de puntos νN generados en P_i . Se generan muestras uniformes en P_i usando HR.

En la implementación proporcionada en la librería VolEsti, para los parámetros de enfriamiento, se establecieron $r = 0,1$ y $\delta = 0,05$ para definir el siguiente cuerpo convexo en MMC con aproximadamente el 10% del volumen del cuerpo anterior, y se estableció que el nivel de significancia sea $\alpha = 0,10$. Por otro lado, se estableció el número de puntos que se generan a partir de P_i en cada paso en $N = 1200 + 2n^2$ y $\nu = 10$. El valor de νN se determinó experimentalmente. Para obtener el punto interior que se necesita para HR, si P es H -politopo el centro de Chebychev.

Apéndice B

Caminatas Aleatorias

B.1. Random-Directions Hit-and-Run:

RDHR ó simplemente Hit-and-Run (HR), es un algoritmo que permite muestrear puntos uniformemente distribuidos dentro de un politopo P [33]. Partiendo de un punto inicial $x_0 \in P$ se selecciona de manera uniforme una dirección e sobre la hiper-esfera d -dimensional, y se buscan los límites de P en esta dirección. Luego seleccionamos un número aleatorio λ , que está uniformemente distribuido entre los límites encontrados.

El nuevo punto muestreado es $x_{k+1} = x_k + \lambda e$

Esta forma de generar los puntos muestreados da como resultado una cadena de Markov que converge hacia la distribución uniforme al aumentar el tamaño de la muestra K . El algoritmo HR converge en tiempo polinómico a una distribución uniforme [27].

Una cadena de Markov es una secuencia correlacionada de estados que cumple que la probabilidad de transición de un estado a otro, solo depende del estado actual. Para garantizar que la distribución sobre un estado de la cadena tiende a $p(x)$ independientemente del estado inicial la cadena debe ser ergódica. Una cadena se dice ergódica si es irreducible y aperiódica. Irreducible significa que se tiene la capacidad de alcanzar cualquier x donde $p(x) > 0$ en un número finito de pasos, y aperiodicidad que no se puede acceder a ningún estado en ciertos momentos regularmente espaciados.

En general, si $f : R^n \rightarrow R^+$ es una función integrable no negativa, entonces define una medida π_f en cualquier subconjunto medible A de R^d ,

$$\pi_f(A) = \frac{\int_A f(x) dx}{\int_{R^d} f(x) dx}$$

Si l es una línea en R^d y $\pi_{l,f}$ es la restricción de π a l ,

$$\pi_{l,f}(P) = \frac{\int_{p+tu \in P} f(p+tu) dt}{\int_l f(x) dx}$$

Donde p es un punto en l y u es un vector unitario paralelo a l . El siguiente pseudocódigo describe un paso de HR.

Algorithm 13: Hit and Run(P, p, f)

$P \subset R^n$;

$p \in P$;

$f : R^n \rightarrow R_+$;

1. seleccionar una línea que pase por p

2. **return** un punto aleatorio en el segmento $l \cap P$ elegido de la distribución $\pi_{l,f}$.

π_f es la distribución estacionaria de HR.

Para encontrar los límites del politopo en la dirección e , buscamos el valor t , que cumple que $x_{k+1} = x_k + te, t > 0$ y t es el mínimo valor por encima del cual se viola una de las restricciones.

Si a_j es el j -ésimo vector fila de A , y b_j el componente j -ésimo de b , este valor puede calcularse a partir de

$$t^{(j)} = \frac{(b_j - a_j x_0)}{a_j e} \quad j = 1..r \quad (\text{B.1})$$

$$t = \min(t^{(1)}, \dots, t^{(r)} \cap t^{(j)} | t^{(j)} > 0)$$

B.2. Coordinate-Directions Hit and Run:

Es una variante del método anterior, pero en lugar de seleccionar una dirección de entre todas las posibles, se selecciona la dirección e , del conjunto acotado de n direcciones fijas $e_1 = (1, 0, \dots, 0)$, $e_2 = (0, 1, 0, \dots)$, \dots , $e_n = (0, 0, \dots, 0, 1)$, encontrando la intersección con los bordes de $K_i [u, v]$ en esta dirección y seleccionando el nuevo punto de manera uniforme dentro del segmento $[u, v]$.

B.3. Ball Walk:

El siguiente pseudocódigo describe un paso de BallWalk:

Algorithm 14: BallWalk(P, p, ∂, f)

$P \subset \mathbb{R}^n$;

$p \in P$;

∂ radio;

$f : \mathbb{R}^n \rightarrow \mathbb{R}_+$;

1. seleccionar de manera uniforme un punto x en la esfera de radio ∂ y centro p
 2. retornar x con probabilidad $\min\{1, \frac{f(x)}{f(p)}\}$ de lo contrario retornar p
-

Donde f es la distribución que van a seguir los puntos muestreados. Si f es la función indicadora de P , entonces la distribución objetivo es la distribución uniforme.

B.4. Billiard Walk:

Billiard Walk es una caminata aleatoria para muestreo de la distribución uniforme. El siguiente pseudocódigo implementa un paso de Billiard Walk, donde $\langle \cdot, \cdot \rangle$ es el producto interno entre dos vectores y $|\cdot|$ es la longitud de un segmento.

Algorithm 15: BilliardWalk(P, p, τ, R)

$P \subset \mathbb{R}^n$;

$p \in P$;

$\tau \in \mathbb{R}_+$;

$R \in \mathbb{N}$;

1. Setear el largo de la trayectoria $L \leftarrow -\tau \ln \eta$, $\eta \sim U(0, 1)$
Setear el numero de rebotes $n \leftarrow 0$ y $p_0 \leftarrow p$
Seleccionar de manera uniforme una dirección v en la esfera de radio uno.
 2. actualizar $n \leftarrow n + 1$; **if** $n > R$ **then** retornar p_0
 3. setear $l \leftarrow \{p + tv, 0 \leq t \leq L\}$
 4. **if** $\partial P \cap l = \emptyset$ **then** retornar $p + Lv$
 5. Actualizar $p \leftarrow \partial P \cap l$; Si s es el vector normal interno del plano tangente en p , $\|s\| = 1$; Actualizar $L \leftarrow L - |P \cap l|$, $v \leftarrow v - 2 \langle v, s \rangle s$; volver a 2
-

Apéndice C

Modelos básicos

En este trabajo vamos a utilizar dos conjuntos de modelos. Primero vamos a utilizar la colección de politopos con las cuales fueron probados los software Vinci y Latte, para validar las implementaciones de los métodos de aproximación en base a los volúmenes calculados con estos software. Esta colección consta de varios modelos de diferentes dimensiones agrupados de la siguiente manera:

- cube: hipercubos con -1 y 1 como coordenadas de vértice, en las dimensiones 2 a 14.
- cross: politopos cruzados, los duales de los cubos de arriba, también en las dimensiones 2 a 14. Si $P \subset R^d$ es un politopo de dimensión d tal que $0 \in \text{int}(P)$. El politopo dual P^* a P se define como $P^* = \{x \in R^n | x \cdot y \leq 1 \forall y \in P\}$.
- rh: politopos construidos al azar eligiendo hiperplanos tangentes a una esfera (en los nombres de los archivos son de la forma rh_d_m, donde d representa la dimensión y m el número de hiperplanos).
- cc: cc_8_5, cc_8_6, cc_8_8, cc_8_9, cc_8_10 y cc_8_11, el producto de dos politopos cíclicos con vértices de 5, 6, 8, 9, 10 y 11 cada uno en la dimensión cuatro. La dimensión final es, por lo tanto, ocho.
- ccp: politopos de cinco a siete vértices., escalados por 2 y luego traducidos por 1 para contener el origen en su interior. Entonces sus vértices tienen coordenadas +1 y -1 ahora.
- metric: politopos métricos de 4 y 6 facetas.
- Birkhoff [1] son politopos que cumplen que la suma de los valores de sus filas y columnas es igual a uno.

$$B_n = \left\{ \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nn} \end{pmatrix} \in R^{n^2} : x_{jk} \geq 0, \quad \begin{matrix} \sum_j x_{ij} = 1 \forall 1 \leq i \leq n \\ \sum_k x_{ij} = 1 \forall 1 \leq j \leq n \end{matrix} \right\}$$

cuyos volúmenes exactos estan calculados hasta B_{10}

Apéndice D

Ejemplo HR aplicado al modelo simple:

A continuación mostramos un ejemplo de este algoritmo aplicado al modelo simple presentado en la sección 3.3.1, utilizando como caminata aleatoria el método RDHR.

Modelo inicial:

$$-x - y + z = 0$$

$$0 \leq x \leq 6$$

$$0 \leq y \leq 8$$

$$0 \leq z \leq 10$$

Luego de procesar el modelo (descomposición QR para obtener la matriz W) pasamos a un modelo en dos dimensiones:

$$0 \leq -0,5774x + 0,5774y \leq 6$$

$$0 \leq -0,7887x + 0,2113y \leq 8$$

$$0 \leq -0,2113x + 0,7887y \leq 10$$

Primero, determinamos el espacio de referencia (figura D.1a) y el punto inicial x_0 resolviendo los problemas de programación lineal.

- $-2,1957 \leq x \leq 8$
- $0 \leq y \leq 12,1957$
- $vol = (x^{max} - x^{min})(y^{max} - y^{min}) = 124,3538$
- $x_0 = (2,9019, 8,0980)$

Iteración 1 (figura D.1b): Muestreo en el espacio de referencia. Se verifica cuantos puntos cumplen con la restricción x_6 (puntos amarillos en la imagen) $n = 10^6, \hat{p}_1 = 0,9313$

Iteración 2 (figura D.1c): Muestreo en el espacio de referencia con restricciones: $x \leq 6$. Se verifica cuantos puntos cumplen con las restricciones: $x \geq 0$. $n = 10^6, \hat{p}_1 = 0,9807$

Iteración 3 (figura D.1d): Muestreo en el espacio de referencia con restricciones: $x \leq 6, x \geq 0$. Se verifica cuantos puntos cumplen con las restricciones: $y \leq 8$. $n = 10^6, \hat{p}_1 = 0,9681$

Iteración 4 (figura D.1e): Muestreo en el espacio de referencia con restricciones: $x \leq 6, x \geq 0, y \leq 8$. Se verifica cuantos puntos cumplen con las restricciones: $y \geq 0$. $n = 10^6, \hat{p}_1 = 0,7047$

Iteración 5 (figura D.1f): Muestreo en el espacio de referencia con restricciones: $x \leq 6, x \geq 0, y \leq 8, y \geq 0$. Se verifica cuantos puntos cumplen con las restricciones: $x + y \leq 10$. $n = 10^6, \hat{p}_1 = 0,8894$

Volumen estimado: 68.91385

Volumen exacto: 69.28203

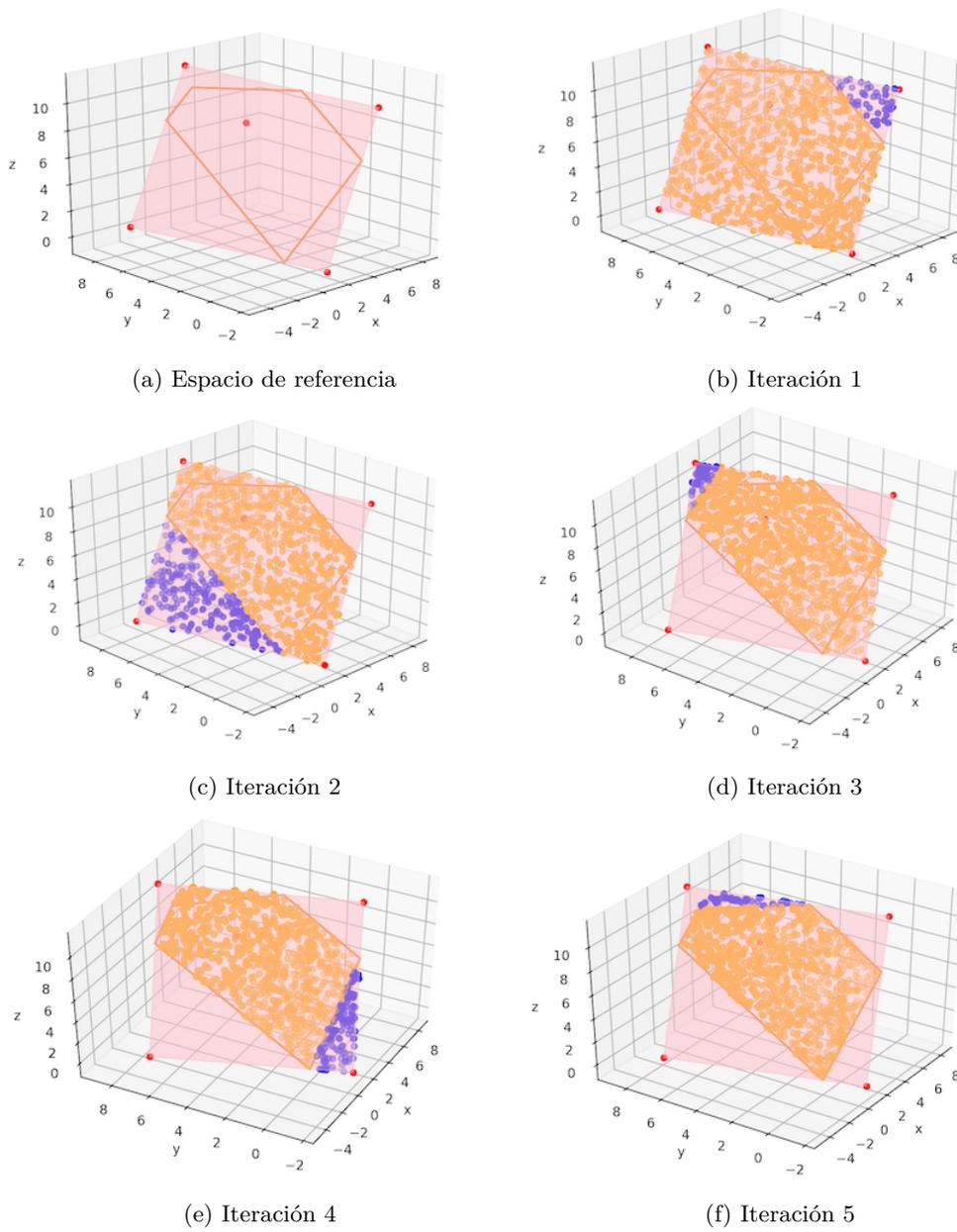


Figura D.1: Cálculo del volumen con HR.

Apéndice E

Ejemplo restricciones con igualdades

Podemos aplicar estas ideas en el caso del modelo simple debido a que este se encuentra definido en R^3 y el espacio de soluciones del mismo es en R^2 .

$$-x - y + z = 0$$

$$0 \leq x \leq 6$$

$$0 \leq y \leq 8$$

$$0 \leq z \leq 10$$

Forma compacta:

$$z = x + y$$

$$0 \leq x \leq 6$$

$$0 \leq y \leq 8$$

$$0 \leq x + y \leq 10$$

$$A_s = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$x_s = (x, y)^T$$

$$b_s = (6, 8, 10, 0, 0)^T$$

$$A_p = (11)$$

$$x_p = z = (11)(x, y)^T = x + y$$

Como en este caso, el espacio de soluciones es un plano, podemos calcular su “volumen” (es decir el área) utilizando la siguiente fórmula:

Área de un triángulo de lados a, b y c: $\sqrt{s(s-a)(s-b)(s-c)}$ con $s = (a + b + c)/2$

Para calcular el área de la parte de soporte la dividimos en tres triángulos.

Área triángulo uno: $\sqrt{64}$

Área triángulo dos: $\sqrt{400}$

Área triángulo tres: $\sqrt{144}$

Área total: 40.0

$$\sqrt{\det(I + A_p^T A_p)} = \sqrt{\det(I + (11)^T (11))} = \sqrt{3}$$

$$Vol(P) = 40 \times \sqrt{3} = 69,2820323$$

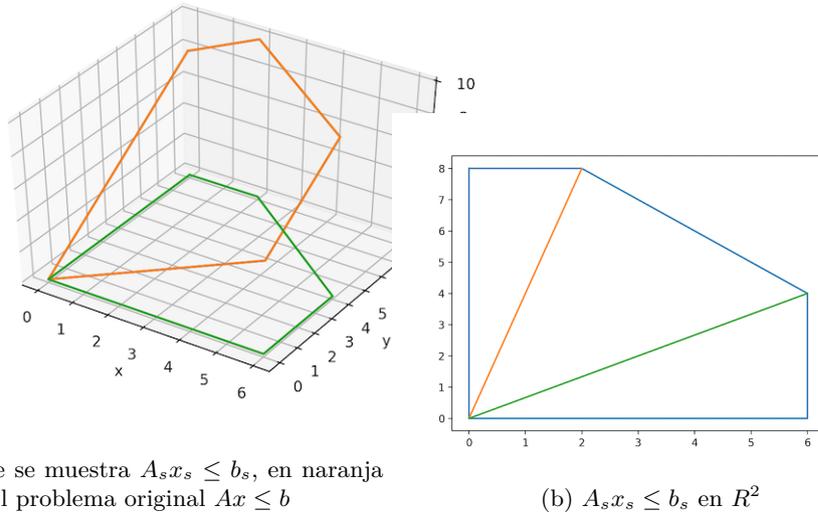


Figura E.1: Ejemplo restricciones de igualdad

Es importante mencionar, que en este caso, como las ecuaciones son igualdades a cero, para encontrar el volumen basta con encontrar una base del espacio nulo ($Nul(A) = x \in R^n : Ax = 0$) y calcular el volumen aplicando las restricciones a dicha base. Cabe destacar que la matriz W especificada en la metodología que se desarrolla en [34] para obtener el espacio de referencia, vendría a ser la matriz M , asociada a la transformación lineal $T : K \rightarrow K'$ que se utiliza en la demostración del teorema 4.5.1.

Apéndice F

Ejemplo Metropolis-Hastings

Supongamos que se quiere sortear muestras a partir de la distribución $p(x)$ representada por la curva azul de la figura F.1a.

Como definimos anteriormente, a medida que vamos construyendo la cadena de Markov vamos calculando $a = \min(1, \frac{p(x)q(x' \leftarrow x)}{p(x')q(x \leftarrow x')})$ para determinar si el nuevo estado propuesto es aceptado o no. En este ejemplo tomamos como función de proposición: $q(x \leftarrow x') = N(x, 1)$ Con esta función de proposición se cumple que $q(x \leftarrow x') = q(x' \leftarrow x)$ por lo tanto $a = \min(1, \frac{p(x')}{p(x)})$

- Paso 1: Supongamos que el punto amarillo en la figura F.1b es el estado inicial, y obtenemos el punto rojo como estado propuesto. Calculamos $a = \min(1, 0,27/0,07) = \min(1, 3,87)$ Como el nuevo estado es más probable que el anterior lo aceptamos. Y la cadena pasa a estar como se muestra en la figura F.1c.
- Paso 2: El punto rojo de la figura F.1c es el estado nuevo propuesto por q . $a = \min(1, 0,28/0,27) = \min(1, 1,01)$. Igual que en el caso anterior el nuevo estado es más probable que el anterior por lo que se acepta. Y la cadena pasa a estar como se muestra en la figura F.1d.
- Paso 3: El punto rojo de la figura F.1d es el estado nuevo propuesto por q . $a = \min(1, 0,04/0,28) = \min(1, 0,13)$. Como el nuevo estado es menos probable que el anterior, lo aceptamos con probabilidad 0.13. Supongamos entonces que la propuesta es rechazada, por lo tanto el nuevo estado es igual al estado anterior. Y la cadena pasa a estar como se muestra en la figura F.1e.
- Paso 4: El punto rojo de la figura F.1e es el estado nuevo propuesto por q . $a = \min(1, 0,20/0,28) = \min(1, 0,73)$. Como el nuevo paso es menos probable que el anterior, lo aceptamos con probabilidad 0.73. Supongamos entonces que la propuesta es rechazada, entonces el nuevo estado es igual al estado anterior. Y la cadena pasa a estar como se muestra en la figura F.1g.

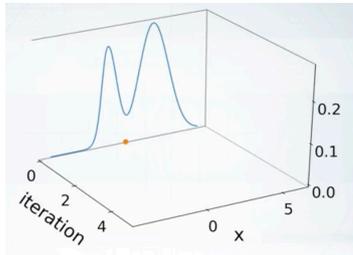
Finalmente luego de 50 iteraciones se obtendrá una cadena como la de la figura F.1h

Si se traza un histograma sobre los puntos obtenidos (figura F.2a), se puede ver que el histograma no es exactamente como la función, pero está cerca, por lo que es una forma razonable de tomar muestras de la curva azul en este caso particular.

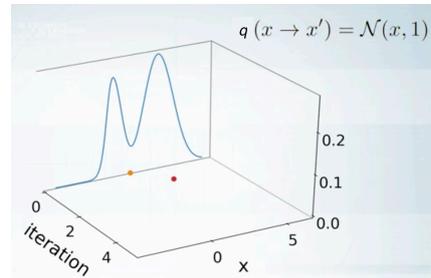
Utilizando este mismo ejemplo podemos ver que variando el tamaño de los pasos de la distribución de propuesta, se puede afectar el desempeño del algoritmo.

Si disminuimos la varianza en la distribución de propuesta q (figura F.2b), estaríamos proponiendo movernos siempre con pasos pequeños alrededor del punto anterior. En cierto modo funciona, pero en la cadena demora en moverse fuera de la región de baja densidad donde comenzó. Lo que significa que la cadena va a demorar más en converger a la distribución deseada.

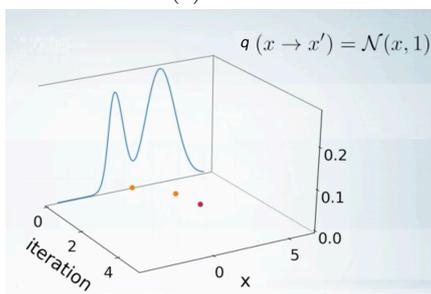
Si incrementamos la varianza en la distribución de propuesta q ((figura F.2c)), vamos a estar dando pasos grandes, esto es mejor en términos de convergencia y de muestras independientes, con respecto a los pasos pequeños. Pero de todas maneras la cadena va a pasar más tiempo en los estados con mayor probabilidad y esto significa que se desperdician recursos y capacidad de cómputo.



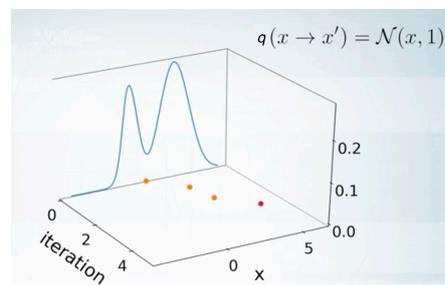
(a) Estado inicial



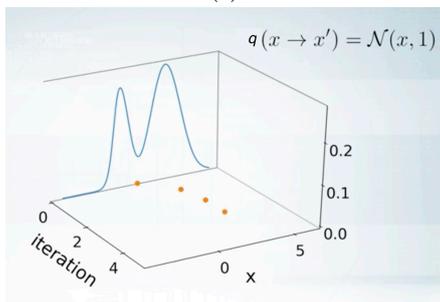
(b) Paso 1



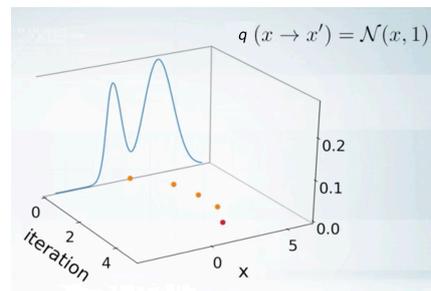
(c) Paso 2



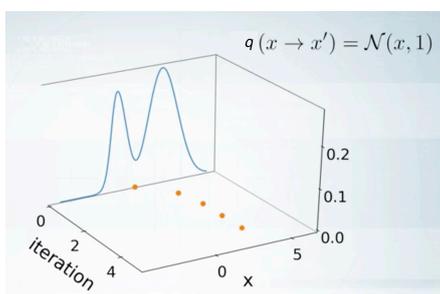
(d) Paso 3



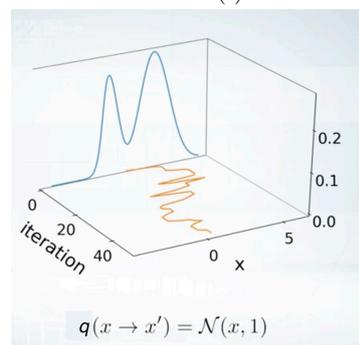
(e) Paso 4



(f) Paso 5

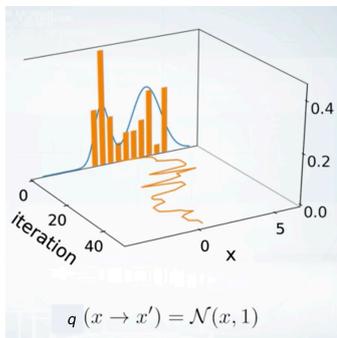


(g) Paso 5



(h) Paso 6

Figura F.1: Ejemplo MH



(a) Varianza 1

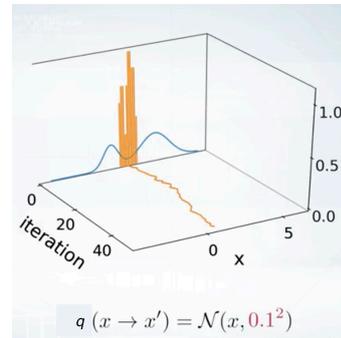
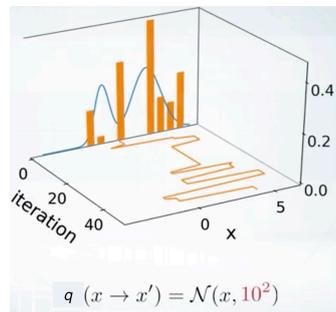
(b) Varianza $0,1^2$ (c) Varianza 10^2

Figura F.2: Ejemplo MH

Apéndice G

Algoritmos implementados

G.1. HR

Algorithm 16: Volumen P basado en HR

Entradas:

K = cantidad de muestras HR ;

$rest_i, i = 1..m$ // Restricciones que definen a P

x^{min}, x^{max} = resultados problema de optimizacion espacio referencia;

$volumen\ espacio\ de\ referencia = (x_1^{max} - x_1^{min}) \dots (x_n^{max} - x_n^{min})$;

$A = [x^{min} \leq I \leq x^{max}]$;

\hat{V} = volumen espacio de referencia;

for $i = 2 \dots m$ **do**

$x_0 \in P$;

 muestras = sortear K muestras dentro de A ;

 muestras positivas = $x \in muestras \wedge x \in A \cup rest_i$;

$\hat{V} = \hat{V} \times (\#muestras\ positivas / K)$;

$A = A \cup [rest_i]$;

end

return \hat{V} ;

G.2. FC

G.2.1. Centro Fijo

Algorithm 17: Volumen FC. Centro fijo

Entradas: $x_0 \in P$;

$S = 0$;

for $i = 1 \dots K$ **do**

 1. Sortear la dirección e , con distribución uniforme

 2. Encontrar todos los puntos donde e se intersecta con P . (En este caso como estamos trabajando con politopos regulares, tenemos una intersección en cada sentido de e , estas dos intersecciones podemos calcular las juntas y hacer dos pasos en uno.)

 3. Calculamos la distancia de x_0 a las dos intersecciones y las acumulamos en S .

 Estimamos el volumen mediante la ecuación: $\hat{V} = E[K\lambda(Z)^n]$

end

G.2.2. Centro Variable

Algorithm 18: Volumen FC. Centro variable

Entradas: $x_0 \in P$;

$S = 0$;

for $i = 1 \dots K$ **do**

1. generar una dirección aleatoria con centro x_{i-1} siguiendo una distribución uniforme.
2. Encontrar la intersección de la nueva dirección en ambos sentidos, calcular la distancia de x_{i-1} a estas dos intersecciones y acumularlas en S.
3. Sortear un número aleatorio λ , que está uniformemente distribuido entre los límites encontrados.
4. $x_i = x_{i-1} + \lambda e$

Estimamos el volumen mediante la ecuación: $\hat{V} = E[K\lambda(Z)^n]$

end

G.3. MCMC

- $f(x) = \frac{\lambda^d(x)}{d}$

- $p_\beta = \frac{1}{Z_\beta} \exp(\beta \ln(f(x)))$

- En la implementación de MH se utiliza como propuesta un paso aleatorio dentro de la esfera de la unidad.

- Para el estimador de volumen, se utiliza $g = \ln(f(x))$.

Algorithm 19: Volumen P basado en MCMC

Entradas:

 $x_0 \in P$; K = cantidad de iteraciones MH ; b = cantidad de cadenas (una por cada p_β); $(\beta_1, \beta_2, \dots, \beta_{b-1}, \beta_b) = (0, 1/b, 2/b, \dots, b-1/b, 1)$ valor de β para cada cadena; $x = x_0$;**for** $n = 1 \dots K$ **do** **for** $j = 1 \dots b$ **do**

Paso MH:

 Sortear un vector e , uniformemente distribuido en la esfera de radio uno. $e = (e_{k-1} + \delta \xi e)$ donde $0 < \delta < 1$ es una constante, y $\xi \in U(0, 1)$ $x' = \frac{e}{\|e\|}$ (con esta distribucion de propuesta se cumple que $q(x \leftarrow x') = q(x' \leftarrow x) = 1$) Calcular $a = \ln\left(\frac{p(x')}{p(x)}\right) = \ln\left(\frac{\exp(\beta_j \ln(f(x')))}{\exp(\beta_j \ln(f(x)))}\right) = \beta_j \ln(f(x')) - \beta_j \ln(f(x))$ **if** $a < \ln(U(0, 1))$ **then** $x = x'$; **else** $x = x$ **end** **end** **for** $k = 1 \dots b$ **do**

Paso PL:

 $x_i \leftarrow U(1, b)$ $x_j \leftarrow U(1, b)$; $x_i \neq x_j$ $\alpha = \min\{1, \exp((\beta_j - \beta_i)(g(x_i) - g(x_j)))\}$ **if** $\alpha < U(0, 1)$ **then** $x_i \leftarrow x_j$ $x_j \leftarrow x_i$ **end** **end** **for** $k = 1 \dots b$ **do** $S_k = S_k + \beta_k \ln(f(x_k))$ **end****end** $\hat{E}_k(\beta_k) = S_k/K \forall \beta \in (\beta_1, \beta_2, \dots, \beta_{b-1}, \beta_b)$ $\hat{V} = Z_0 \exp(\Delta \beta (\frac{\hat{E}_0 + \hat{E}_1}{2} + \sum_{j=1}^{n-1} \hat{E}_j \Delta \beta))$

G.4. Resultados computacionales

En las secciones 5.5 y 5.6 se aplicó el método Bland-Altman para analizar los resultados obtenidos al utilizar los algoritmos CG y CB para aproximar el volumen de los modelos nucleo y compacto de E. coli. A continuación presentamos los resultados tabulados que permitieron realizar dicho análisis.

G.4.1. Resultados Modelo compacto E. coli CG y CB

| P | d | N | CB μ | CG μ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----|-----|-------|----------|----------|-----------------------|---------------------------|
| V08 | 6 | 10000 | 1.44e+09 | 1.45e+09 | 0.01 | 2.11e+01 |
| V09 | 6 | 10000 | 1.71e+09 | 1.69e+09 | -0.01 | 2.13e+01 |
| V10 | 6 | 10000 | 3.52e+09 | 3.76e+09 | 0.07 | 2.20e+01 |
| V11 | 6 | 10000 | 2.49e+09 | 2.61e+09 | 0.05 | 2.17e+01 |
| V12 | 6 | 10000 | 1.74e+09 | 1.74e+09 | 0.00 | 2.13e+01 |
| V13 | 6 | 10000 | 1.57e+09 | 1.41e+09 | -0.10 | 2.11e+01 |
| V17 | 6 | 10000 | 2.22e+09 | 2.05e+09 | -0.08 | 2.15e+01 |

Tabla G.1: Resultados modelo compacto E.coli con una fermentación prendida.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|--------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09 | 7 | 10000 | 8.64e+09 | 9.60e+09 | 0.11 | 2.29e+01 |
| V08V10 | 7 | 10000 | 1.70e+10 | 1.81e+10 | 0.06 | 2.36e+01 |
| V08V11 | 7 | 10000 | 1.56e+10 | 1.50e+10 | -0.04 | 2.34e+01 |
| V08V12 | 7 | 10000 | 9.11e+09 | 8.78e+09 | -0.04 | 2.29e+01 |
| V08V13 | 7 | 10000 | 8.07e+09 | 8.43e+09 | 0.04 | 2.28e+01 |
| V08V17 | 7 | 10000 | 1.16e+10 | 1.15e+10 | -0.01 | 2.32e+01 |
| V09V10 | 7 | 10000 | 1.91e+10 | 1.94e+10 | 0.01 | 2.37e+01 |
| V09V11 | 7 | 10000 | 2.40e+10 | 2.31e+10 | -0.04 | 2.39e+01 |
| V09V12 | 7 | 10000 | 1.09e+10 | 1.18e+10 | 0.08 | 2.32e+01 |
| V09V13 | 7 | 10000 | 9.65e+09 | 9.54e+09 | -0.01 | 2.30e+01 |
| V09V17 | 7 | 10000 | 1.45e+10 | 1.46e+10 | 0.01 | 2.34e+01 |
| V10V11 | 7 | 10000 | 5.19e+10 | 5.19e+10 | 0.00 | 2.47e+01 |
| V10V12 | 7 | 10000 | 1.93e+10 | 1.97e+10 | 0.02 | 2.37e+01 |
| V10V13 | 7 | 10000 | 1.72e+10 | 1.79e+10 | 0.04 | 2.36e+01 |
| V10V17 | 7 | 10000 | 2.86e+10 | 2.84e+10 | -0.01 | 2.41e+01 |
| V11V12 | 7 | 10000 | 2.29e+10 | 2.17e+10 | -0.05 | 2.38e+01 |
| V11V13 | 7 | 10000 | 1.92e+10 | 1.78e+10 | -0.08 | 2.36e+01 |
| V11V17 | 7 | 10000 | 2.71e+10 | 2.73e+10 | 0.01 | 2.40e+01 |
| V12V13 | 7 | 10000 | 7.47e+09 | 7.33e+09 | -0.02 | 2.27e+01 |
| V12V17 | 7 | 10000 | 9.99e+09 | 9.49e+09 | -0.05 | 2.30e+01 |
| V13V17 | 7 | 10000 | 9.07e+09 | 8.82e+09 | -0.03 | 2.29e+01 |

Tabla G.2: Resultados modelo compacto E.coli con dos fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09V10 | 8 | 10000 | 7.75e+10 | 7.52e+10 | -0.03 | 2.51e+01 |
| V08V09V11 | 8 | 10000 | 1.07e+11 | 1.09e+11 | 0.02 | 2.54e+01 |
| V08V09V12 | 8 | 10000 | 4.11e+10 | 4.66e+10 | 0.13 | 2.45e+01 |
| V08V09V13 | 8 | 10000 | 3.74e+10 | 3.93e+10 | 0.05 | 2.44e+01 |
| V08V09V17 | 8 | 10000 | 6.21e+10 | 5.66e+10 | -0.09 | 2.48e+01 |
| V08V10V11 | 8 | 10000 | 2.01e+11 | 2.07e+11 | 0.03 | 2.60e+01 |
| V08V10V12 | 8 | 10000 | 8.42e+10 | 7.98e+10 | -0.05 | 2.51e+01 |
| V08V10V13 | 8 | 10000 | 6.90e+10 | 7.08e+10 | 0.03 | 2.50e+01 |
| V08V10V17 | 8 | 10000 | 1.17e+11 | 1.09e+11 | -0.07 | 2.55e+01 |
| V08V11V12 | 8 | 10000 | 8.73e+10 | 9.61e+10 | 0.10 | 2.52e+01 |
| V08V11V13 | 8 | 10000 | 8.28e+10 | 8.37e+10 | 0.01 | 2.51e+01 |
| V08V11V17 | 8 | 10000 | 1.16e+11 | 1.14e+11 | -0.01 | 2.55e+01 |
| V08V12V13 | 8 | 10000 | 2.86e+10 | 3.11e+10 | 0.08 | 2.41e+01 |
| V08V12V17 | 8 | 10000 | 4.35e+10 | 4.46e+10 | 0.02 | 2.45e+01 |
| V08V13V17 | 8 | 10000 | 4.32e+10 | 4.08e+10 | -0.06 | 2.45e+01 |
| V09V10V11 | 8 | 10000 | 2.45e+11 | 2.59e+11 | 0.06 | 2.63e+01 |
| V09V10V12 | 8 | 10000 | 9.04e+10 | 9.00e+10 | -0.00 | 2.52e+01 |
| V09V10V13 | 8 | 10000 | 8.13e+10 | 8.11e+10 | -0.00 | 2.51e+01 |
| V09V10V17 | 8 | 10000 | 1.23e+11 | 1.21e+11 | -0.02 | 2.55e+01 |
| V09V11V12 | 8 | 10000 | 1.33e+11 | 1.33e+11 | -0.00 | 2.56e+01 |
| V09V11V13 | 8 | 10000 | 1.13e+11 | 1.15e+11 | 0.01 | 2.55e+01 |
| V09V11V17 | 8 | 10000 | 1.64e+11 | 1.67e+11 | 0.02 | 2.58e+01 |
| V09V12V13 | 8 | 10000 | 3.64e+10 | 3.55e+10 | -0.03 | 2.43e+01 |
| V09V12V17 | 8 | 10000 | 5.52e+10 | 5.18e+10 | -0.06 | 2.47e+01 |
| V09V13V17 | 8 | 10000 | 4.43e+10 | 4.94e+10 | 0.11 | 2.46e+01 |
| V10V11V12 | 8 | 10000 | 2.52e+11 | 2.47e+11 | -0.02 | 2.62e+01 |
| V10V11V13 | 8 | 10000 | 2.11e+11 | 2.29e+11 | 0.08 | 2.61e+01 |
| V10V11V17 | 8 | 10000 | 3.26e+11 | 3.22e+11 | -0.01 | 2.65e+01 |
| V10V12V13 | 8 | 10000 | 5.79e+10 | 6.34e+10 | 0.09 | 2.48e+01 |
| V10V12V17 | 8 | 10000 | 8.50e+10 | 9.41e+10 | 0.10 | 2.52e+01 |
| V10V13V17 | 8 | 10000 | 8.14e+10 | 8.45e+10 | 0.04 | 2.51e+01 |
| V11V12V13 | 8 | 10000 | 8.30e+10 | 8.01e+10 | -0.04 | 2.51e+01 |
| V11V12V17 | 8 | 10000 | 1.13e+11 | 1.13e+11 | 0.00 | 2.55e+01 |
| V11V13V17 | 8 | 10000 | 1.08e+11 | 1.05e+11 | -0.02 | 2.54e+01 |
| V12V13V17 | 8 | 10000 | 2.93e+10 | 3.01e+10 | 0.03 | 2.41e+01 |

Tabla G.3: Resultados modelo compacto E.coli con tres fermentaciones prendidas.

| <i>P</i> | <i>d</i> | <i>N</i> | <i>CB</i> μ | <i>CG</i> μ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|--------------|----------|----------|-----------------|-----------------|-----------------------|---------------------------|
| V08V09V10V11 | 9 | 10000 | 8.77e+11 | 9.05e+11 | 0.03 | 2.75e+01 |
| V08V09V10V12 | 9 | 10000 | 2.81e+11 | 2.82e+11 | 0.00 | 2.64e+01 |
| V08V09V10V13 | 9 | 10000 | 2.72e+11 | 3.20e+11 | 0.16 | 2.64e+01 |
| V08V09V10V17 | 9 | 10000 | 4.01e+11 | 4.10e+11 | 0.02 | 2.67e+01 |
| V08V09V11V12 | 9 | 10000 | 4.44e+11 | 4.45e+11 | 0.00 | 2.68e+01 |
| V08V09V11V13 | 9 | 10000 | 3.89e+11 | 4.28e+11 | 0.10 | 2.67e+01 |
| V08V09V11V17 | 9 | 10000 | 5.65e+11 | 5.26e+11 | -0.07 | 2.70e+01 |
| V08V09V12V13 | 9 | 10000 | 1.29e+11 | 1.26e+11 | -0.03 | 2.56e+01 |
| V08V09V12V17 | 9 | 10000 | 1.78e+11 | 1.68e+11 | -0.06 | 2.59e+01 |
| V08V09V13V17 | 9 | 10000 | 1.75e+11 | 1.65e+11 | -0.06 | 2.59e+01 |
| V08V10V11V12 | 9 | 10000 | 8.90e+11 | 8.97e+11 | 0.01 | 2.75e+01 |
| V08V10V11V13 | 9 | 10000 | 7.14e+11 | 7.77e+11 | 0.08 | 2.73e+01 |
| V08V10V11V17 | 9 | 10000 | 1.14e+12 | 1.12e+12 | -0.01 | 2.78e+01 |
| V08V10V12V13 | 9 | 10000 | 1.91e+11 | 2.12e+11 | 0.11 | 2.60e+01 |
| V08V10V12V17 | 9 | 10000 | 3.04e+11 | 3.01e+11 | -0.01 | 2.64e+01 |
| V08V10V13V17 | 9 | 10000 | 3.09e+11 | 2.88e+11 | -0.07 | 2.64e+01 |
| V08V11V12V13 | 9 | 10000 | 3.46e+11 | 2.90e+11 | -0.18 | 2.65e+01 |
| V08V11V12V17 | 9 | 10000 | 4.14e+11 | 4.29e+11 | 0.03 | 2.68e+01 |
| V08V11V13V17 | 9 | 10000 | 3.95e+11 | 4.09e+11 | 0.03 | 2.67e+01 |
| V08V12V13V17 | 9 | 10000 | 1.10e+11 | 1.01e+11 | -0.08 | 2.54e+01 |
| V09V10V11V12 | 9 | 10000 | 1.09e+12 | 1.09e+12 | 0.00 | 2.77e+01 |
| V09V10V11V13 | 9 | 10000 | 1.04e+12 | 1.05e+12 | 0.01 | 2.77e+01 |
| V09V10V11V17 | 9 | 10000 | 1.19e+12 | 1.43e+12 | 0.18 | 2.79e+01 |
| V09V10V12V13 | 9 | 10000 | 2.43e+11 | 2.19e+11 | -0.10 | 2.62e+01 |
| V09V10V12V17 | 9 | 10000 | 3.61e+11 | 3.50e+11 | -0.03 | 2.66e+01 |
| V09V10V13V17 | 9 | 10000 | 3.32e+11 | 3.22e+11 | -0.03 | 2.65e+01 |
| V09V11V12V13 | 9 | 10000 | 4.28e+11 | 3.97e+11 | -0.07 | 2.67e+01 |
| V09V11V12V17 | 9 | 10000 | 5.39e+11 | 6.18e+11 | 0.14 | 2.71e+01 |
| V09V11V13V17 | 9 | 10000 | 5.63e+11 | 5.35e+11 | -0.05 | 2.70e+01 |
| V09V12V13V17 | 9 | 10000 | 1.53e+11 | 1.48e+11 | -0.03 | 2.57e+01 |
| V10V11V12V13 | 9 | 10000 | 6.67e+11 | 7.44e+11 | 0.11 | 2.73e+01 |
| V10V11V12V17 | 9 | 10000 | 1.08e+12 | 1.11e+12 | 0.03 | 2.77e+01 |
| V10V11V13V17 | 9 | 10000 | 1.19e+12 | 1.10e+12 | -0.08 | 2.78e+01 |
| V10V12V13V17 | 9 | 10000 | 1.99e+11 | 2.37e+11 | 0.17 | 2.61e+01 |
| V11V12V13V17 | 9 | 10000 | 3.56e+11 | 3.27e+11 | -0.09 | 2.66e+01 |

Tabla G.4: Resultados modelo compacto E.coli con cuatro fermentaciones prendidas.

| <i>P</i> | <i>d</i> | <i>N</i> | <i>CB</i> μ | <i>CG</i> μ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----------------|----------|----------|-----------------|-----------------|-----------------------|---------------------------|
| V08V09V10V11V12 | 10 | 10000 | 3.20e+12 | 3.12e+12 | -0.03 | 2.88e+01 |
| V08V09V10V11V13 | 10 | 10000 | 2.69e+12 | 2.72e+12 | 0.01 | 2.86e+01 |
| V08V09V10V11V17 | 10 | 10000 | 3.81e+12 | 4.25e+12 | 0.11 | 2.90e+01 |
| V08V09V10V12V13 | 10 | 10000 | 6.11e+11 | 6.74e+11 | 0.10 | 2.72e+01 |
| V08V09V10V12V17 | 10 | 10000 | 1.03e+12 | 1.12e+12 | 0.09 | 2.77e+01 |
| V08V09V10V13V17 | 10 | 10000 | 1.08e+12 | 1.30e+12 | 0.18 | 2.78e+01 |
| V08V09V11V12V13 | 10 | 10000 | 1.29e+12 | 1.19e+12 | -0.08 | 2.78e+01 |
| V08V09V11V12V17 | 10 | 10000 | 1.71e+12 | 1.78e+12 | 0.04 | 2.82e+01 |
| V08V09V11V13V17 | 10 | 10000 | 1.64e+12 | 1.52e+12 | -0.08 | 2.81e+01 |
| V08V09V12V13V17 | 10 | 10000 | 4.06e+11 | 4.25e+11 | 0.05 | 2.68e+01 |
| V08V10V11V12V13 | 10 | 10000 | 2.14e+12 | 2.35e+12 | 0.10 | 2.84e+01 |
| V08V10V11V12V17 | 10 | 10000 | 3.15e+12 | 3.03e+12 | -0.04 | 2.88e+01 |
| V08V10V11V13V17 | 10 | 10000 | 2.78e+12 | 2.95e+12 | 0.06 | 2.87e+01 |
| V08V10V12V13V17 | 10 | 10000 | 6.46e+11 | 6.60e+11 | 0.02 | 2.72e+01 |
| V08V11V12V13V17 | 10 | 10000 | 1.06e+12 | 1.07e+12 | 0.01 | 2.77e+01 |
| V09V10V11V12V13 | 10 | 10000 | 2.69e+12 | 2.51e+12 | -0.07 | 2.86e+01 |
| V09V10V11V12V17 | 10 | 10000 | 3.74e+12 | 3.90e+12 | 0.04 | 2.90e+01 |
| V09V10V11V13V17 | 10 | 10000 | 4.10e+12 | 3.83e+12 | -0.07 | 2.90e+01 |
| V09V10V12V13V17 | 10 | 10000 | 7.01e+11 | 7.85e+11 | 0.11 | 2.73e+01 |
| V09V11V12V13V17 | 10 | 10000 | 1.48e+12 | 1.41e+12 | -0.05 | 2.80e+01 |
| V10V11V12V13V17 | 10 | 10000 | 2.35e+12 | 2.46e+12 | 0.05 | 2.85e+01 |

Tabla G.5: Resultados modelo compacto E.coli con cinco fermentaciones prendidas.

| <i>P</i> | <i>d</i> | <i>N</i> | <i>CB</i> μ | <i>CG</i> μ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|--------------------|----------|----------|-----------------|-----------------|-----------------------|---------------------------|
| V08V09V10V11V12V13 | 11 | 10000 | 6.54e+12 | 6.88e+12 | 0.05 | 2.95e+01 |
| V08V09V10V11V12V17 | 11 | 10000 | 1.21e+13 | 1.12e+13 | -0.08 | 3.01e+01 |
| V08V09V10V11V13V17 | 11 | 10000 | 9.34e+12 | 9.27e+12 | -0.01 | 2.99e+01 |
| V08V09V10V12V13V17 | 11 | 10000 | 2.00e+12 | 2.05e+12 | 0.03 | 2.83e+01 |
| V08V09V11V12V13V17 | 11 | 10000 | 3.96e+12 | 3.64e+12 | -0.08 | 2.90e+01 |
| V08V10V11V12V13V17 | 11 | 10000 | 6.00e+12 | 6.37e+12 | 0.06 | 2.95e+01 |
| V09V10V11V12V13V17 | 11 | 10000 | 8.19e+12 | 8.31e+12 | 0.01 | 2.97e+01 |

Tabla G.6: Resultados modelo compacto E.coli con seis fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----------------------|-----|----------|----------|----------|-----------------------|---------------------------|
| V08V09V10V11V12V13V17 | 12 | 1.00e+04 | 4.42e+14 | 5.20e+14 | 0.16 | 3.38e+01 |

Tabla G.7: Resultados modelo compacto E.coli con todas las fermentaciones prendidas.

G.4.2. Resultados Modelo núcleo E. coli CG y CB

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----|-----|-------|----------|----------|-----------------------|---------------------------|
| V08 | 21 | 10000 | 2.92e+49 | 2.54e+49 | -0.14 | 1.14e+02 |
| V09 | 22 | 10000 | 4.95e+51 | 2.70e+55 | 8.60 | 1.23e+02 |
| V10 | 21 | 10000 | 1.59e+48 | 1.81e+48 | 0.13 | 1.11e+02 |
| V11 | 22 | 10000 | 1.44e+50 | 1.30e+50 | -0.10 | 1.15e+02 |
| V12 | 21 | 10000 | 1.11e+48 | 1.44e+48 | 0.26 | 1.11e+02 |
| V13 | 21 | 10000 | 6.15e+48 | 8.00e+48 | 0.26 | 1.12e+02 |
| V17 | 21 | 10000 | 4.87e+51 | 4.14e+51 | -0.16 | 1.19e+02 |

Tabla G.8: Resultados modelo núcleo E.coli con una fermentación prendida.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|--------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09 | 23 | 10000 | 5.56e+54 | 1.63e+55 | 1.08 | 1.27e+02 |
| V08V10 | 22 | 10000 | 7.74e+51 | 9.29e+51 | 0.18 | 1.20e+02 |
| V08V11 | 23 | 10000 | 1.55e+53 | 5.80e+53 | 1.32 | 1.23e+02 |
| V08V12 | 22 | 10000 | 1.37e+52 | 6.37e+52 | 1.54 | 1.21e+02 |
| V08V13 | 22 | 10000 | 3.01e+52 | 6.63e+52 | 0.79 | 1.21e+02 |
| V08V17 | 22 | 10000 | 3.45e+55 | 4.41e+55 | 0.25 | 1.28e+02 |
| V09V10 | 23 | 10000 | 4.80e+54 | 5.72e+54 | 0.17 | 1.26e+02 |
| V09V11 | 24 | 10000 | 2.70e+57 | 2.74e+57 | 0.01 | 1.32e+02 |
| V09V12 | 23 | 10000 | 7.54e+54 | 5.80e+54 | -0.26 | 1.26e+02 |
| V09V13 | 23 | 10000 | 1.82e+55 | 2.41e+55 | 0.28 | 1.27e+02 |
| V09V17 | 23 | 10000 | 4.77e+58 | 4.63e+58 | -0.03 | 1.35e+02 |
| V10V11 | 23 | 10000 | 5.56e+53 | 7.36e+53 | 0.28 | 1.24e+02 |
| V10V12 | 22 | 10000 | 3.01e+50 | 6.65e+50 | 0.79 | 1.17e+02 |
| V10V13 | 22 | 10000 | 3.32e+51 | 2.73e+51 | -0.20 | 1.19e+02 |
| V10V17 | 22 | 10000 | 2.29e+54 | 1.40e+55 | 1.81 | 1.26e+02 |
| V11V12 | 23 | 10000 | 4.77e+52 | 1.17e+53 | 0.90 | 1.22e+02 |
| V11V13 | 23 | 10000 | 3.96e+53 | 9.07e+53 | 0.83 | 1.24e+02 |
| V11V17 | 23 | 10000 | 2.01e+56 | 4.47e+56 | 0.80 | 1.30e+02 |
| V12V13 | 22 | 10000 | 3.51e+52 | 1.92e+52 | -0.60 | 1.21e+02 |
| V12V17 | 22 | 10000 | 2.30e+54 | 2.40e+54 | 0.04 | 1.25e+02 |
| V13V17 | 22 | 10000 | 2.44e+54 | 2.76e+54 | 0.12 | 1.25e+02 |

Tabla G.9: Resultados modelo núcleo E.coli con dos fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09V10 | 24 | 10000 | 8.76e+57 | 1.48e+58 | 0.52 | 1.34e+02 |
| V08V09V11 | 25 | 10000 | 2.33e+60 | 3.07e+60 | 0.28 | 1.39e+02 |
| V08V09V12 | 24 | 10000 | 1.15e+58 | 1.79e+58 | 0.44 | 1.34e+02 |
| V08V09V13 | 24 | 10000 | 3.03e+58 | 4.65e+58 | 0.43 | 1.35e+02 |
| V08V09V17 | 24 | 10000 | 4.68e+61 | 9.63e+61 | 0.72 | 1.42e+02 |
| V08V10V11 | 24 | 10000 | 1.69e+57 | 2.51e+57 | 0.40 | 1.32e+02 |
| V08V10V12 | 23 | 10000 | 3.81e+54 | 1.35e+58 | 8.17 | 1.30e+02 |
| V08V10V13 | 23 | 10000 | 1.86e+55 | 2.69e+55 | 0.37 | 1.27e+02 |
| V08V10V17 | 23 | 10000 | 3.88e+58 | 3.32e+58 | -0.16 | 1.35e+02 |
| V08V11V12 | 24 | 10000 | 8.06e+56 | 1.62e+57 | 0.70 | 1.31e+02 |
| V08V11V13 | 24 | 10000 | 7.89e+57 | 8.05e+57 | 0.02 | 1.33e+02 |
| V08V11V17 | 24 | 10000 | 6.32e+60 | 7.02e+60 | 0.10 | 1.40e+02 |
| V08V12V13 | 23 | 10000 | 2.42e+55 | 7.08e+55 | 1.07 | 1.28e+02 |
| V08V12V17 | 23 | 10000 | 2.00e+58 | 2.57e+58 | 0.25 | 1.34e+02 |
| V08V13V17 | 23 | 10000 | 1.49e+58 | 3.84e+58 | 0.95 | 1.34e+02 |
| V09V10V11 | 25 | 10000 | 8.63e+59 | 4.22e+60 | 1.59 | 1.39e+02 |
| V09V10V12 | 24 | 10000 | 3.61e+57 | 1.34e+59 | 3.61 | 1.34e+02 |
| V09V10V13 | 24 | 10000 | 1.29e+58 | 2.07e+58 | 0.47 | 1.34e+02 |
| V09V10V17 | 24 | 10000 | 1.79e+61 | 5.38e+61 | 1.10 | 1.42e+02 |
| V09V11V12 | 25 | 10000 | 1.20e+60 | 3.87e+60 | 1.17 | 1.39e+02 |
| V09V11V13 | 25 | 10000 | 8.09e+60 | 1.02e+61 | 0.23 | 1.40e+02 |
| V09V11V17 | 25 | 10000 | 1.86e+64 | 1.25e+64 | -0.40 | 1.48e+02 |
| V09V12V13 | 24 | 10000 | 6.57e+57 | 2.53e+58 | 1.35 | 1.34e+02 |
| V09V12V17 | 24 | 10000 | 1.23e+61 | 2.16e+61 | 0.56 | 1.41e+02 |
| V09V13V17 | 24 | 10000 | 2.74e+61 | 2.69e+61 | -0.02 | 1.41e+02 |
| V10V11V12 | 24 | 10000 | 1.13e+56 | 3.24e+59 | 7.96 | 1.33e+02 |
| V10V11V13 | 24 | 10000 | 1.58e+57 | 1.99e+57 | 0.23 | 1.32e+02 |
| V10V11V17 | 24 | 10000 | 4.27e+59 | 6.34e+59 | 0.40 | 1.38e+02 |
| V10V12V13 | 23 | 10000 | 7.45e+54 | 1.19e+55 | 0.47 | 1.27e+02 |
| V10V12V17 | 23 | 10000 | 1.09e+57 | 1.09e+57 | 0.00 | 1.31e+02 |
| V10V13V17 | 23 | 10000 | 1.16e+57 | 1.25e+57 | 0.07 | 1.31e+02 |
| V11V12V13 | 24 | 10000 | 8.44e+56 | 2.64e+57 | 1.14 | 1.32e+02 |
| V11V12V17 | 24 | 10000 | 3.75e+59 | 2.82e+59 | -0.28 | 1.37e+02 |
| V11V13V17 | 24 | 10000 | 2.55e+59 | 2.53e+59 | -0.01 | 1.37e+02 |
| V12V13V17 | 23 | 10000 | 3.00e+58 | 2.36e+58 | -0.24 | 1.35e+02 |

Tabla G.10: Resultados modelo núcleo E.coli con tres fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|--------------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09V10V11 | 26 | 10000 | 3.25e+63 | 6.37e+63 | 0.67 | 1.47e+02 |
| V08V09V10V12 | 25 | 10000 | 1.12e+61 | 1.60e+61 | 0.36 | 1.41e+02 |
| V08V09V10V13 | 25 | 10000 | 1.66e+61 | 5.08e+61 | 1.12 | 1.42e+02 |
| V08V09V10V17 | 25 | 10000 | 3.52e+64 | 7.42e+64 | 0.75 | 1.49e+02 |
| V08V09V11V12 | 26 | 10000 | 4.55e+63 | 6.24e+63 | 0.32 | 1.47e+02 |
| V08V09V11V13 | 26 | 10000 | 1.37e+64 | 1.77e+64 | 0.26 | 1.48e+02 |
| V08V09V11V17 | 26 | 10000 | 1.56e+67 | 2.01e+67 | 0.25 | 1.55e+02 |
| V08V09V12V13 | 25 | 10000 | 2.19e+61 | 5.70e+61 | 0.96 | 1.42e+02 |
| V08V09V12V17 | 25 | 10000 | 1.68e+64 | 4.01e+64 | 0.87 | 1.48e+02 |
| V08V09V13V17 | 25 | 10000 | 3.30e+64 | 5.37e+64 | 0.49 | 1.49e+02 |
| V08V10V11V12 | 25 | 10000 | 1.68e+60 | 3.03e+60 | 0.59 | 1.39e+02 |
| V08V10V11V13 | 25 | 10000 | 3.95e+60 | 1.07e+61 | 1.00 | 1.40e+02 |
| V08V10V11V17 | 25 | 10000 | 8.78e+63 | 7.85e+63 | -0.11 | 1.47e+02 |
| V08V10V12V13 | 24 | 10000 | 1.52e+58 | 3.60e+58 | 0.86 | 1.34e+02 |
| V08V10V12V17 | 24 | 10000 | 6.35e+60 | 2.01e+61 | 1.15 | 1.41e+02 |
| V08V10V13V17 | 24 | 10000 | 1.78e+61 | 1.96e+61 | 0.10 | 1.41e+02 |
| V08V11V12V13 | 25 | 10000 | 5.68e+60 | 9.37e+60 | 0.50 | 1.40e+02 |
| V08V11V12V17 | 25 | 10000 | 2.37e+63 | 3.40e+63 | 0.36 | 1.46e+02 |
| V08V11V13V17 | 25 | 10000 | 2.34e+63 | 3.48e+63 | 0.40 | 1.46e+02 |
| V08V12V13V17 | 24 | 10000 | 2.98e+61 | 7.49e+62 | 3.22 | 1.43e+02 |
| V09V10V11V12 | 26 | 10000 | 2.34e+63 | 4.58e+63 | 0.67 | 1.46e+02 |
| V09V10V11V13 | 26 | 10000 | 6.91e+63 | 1.69e+64 | 0.89 | 1.47e+02 |
| V09V10V11V17 | 26 | 10000 | 1.51e+67 | 1.58e+67 | 0.05 | 1.55e+02 |
| V09V10V12V13 | 25 | 10000 | 1.09e+61 | 2.98e+61 | 1.01 | 1.41e+02 |
| V09V10V12V17 | 25 | 10000 | 2.91e+64 | 2.16e+64 | -0.30 | 1.48e+02 |
| V09V10V13V17 | 25 | 10000 | 2.54e+64 | 2.72e+64 | 0.07 | 1.48e+02 |
| V09V11V12V13 | 26 | 10000 | 2.54e+63 | 9.86e+63 | 1.36 | 1.47e+02 |
| V09V11V12V17 | 26 | 10000 | 4.06e+66 | 7.24e+66 | 0.58 | 1.54e+02 |
| V09V11V13V17 | 26 | 10000 | 5.65e+66 | 9.28e+66 | 0.50 | 1.54e+02 |
| V09V12V13V17 | 25 | 10000 | 3.96e+64 | 4.79e+63 | -2.11 | 1.48e+02 |
| V10V11V12V13 | 25 | 10000 | 2.35e+60 | 6.75e+60 | 1.06 | 1.40e+02 |
| V10V11V12V17 | 25 | 10000 | 2.73e+62 | 3.09e+62 | 0.12 | 1.44e+02 |
| V10V11V13V17 | 25 | 10000 | 4.52e+62 | 3.55e+62 | -0.24 | 1.44e+02 |
| V10V12V13V17 | 24 | 10000 | 1.29e+61 | 1.31e+61 | 0.02 | 1.41e+02 |
| V11V12V13V17 | 25 | 10000 | 2.17e+63 | 3.44e+63 | 0.46 | 1.46e+02 |

Tabla G.11: Resultados modelo núcleo E.coli con cuatro fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----------------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09V10V11V12 | 27 | 10000 | 3.04e+66 | 8.14e+66 | 0.98 | 1.54e+02 |
| V08V09V10V11V13 | 27 | 10000 | 1.12e+67 | 2.12e+67 | 0.64 | 1.55e+02 |
| V08V09V10V11V17 | 27 | 10000 | 6.35e+69 | 2.44e+70 | 1.35 | 1.61e+02 |
| V08V09V10V12V13 | 26 | 10000 | 2.61e+64 | 6.17e+64 | 0.86 | 1.49e+02 |
| V08V09V10V12V17 | 26 | 10000 | 3.89e+67 | 4.99e+67 | 0.25 | 1.56e+02 |
| V08V09V10V13V17 | 26 | 10000 | 4.45e+67 | 5.98e+67 | 0.30 | 1.56e+02 |
| V08V09V11V12V13 | 27 | 10000 | 1.06e+67 | 1.69e+67 | 0.47 | 1.55e+02 |
| V08V09V11V12V17 | 27 | 10000 | 5.42e+69 | 1.16e+70 | 0.76 | 1.61e+02 |
| V08V09V11V13V17 | 27 | 10000 | 6.21e+69 | 1.47e+70 | 0.86 | 1.61e+02 |
| V08V09V12V13V17 | 26 | 10000 | 6.93e+67 | 8.07e+67 | 0.15 | 1.56e+02 |
| V08V10V11V12V13 | 26 | 10000 | 1.62e+63 | 7.98e+63 | 1.59 | 1.46e+02 |
| V08V10V11V12V17 | 26 | 10000 | 2.68e+66 | 4.12e+66 | 0.43 | 1.53e+02 |
| V08V10V11V13V17 | 26 | 10000 | 5.70e+66 | 3.84e+66 | -0.40 | 1.54e+02 |
| V08V10V12V13V17 | 25 | 10000 | 4.11e+64 | 4.97e+64 | 0.19 | 1.49e+02 |
| V08V11V12V13V17 | 26 | 10000 | 3.72e+66 | 8.19e+66 | 0.79 | 1.54e+02 |
| V09V10V11V12V13 | 27 | 10000 | 3.47e+66 | 3.67e+67 | 2.36 | 1.54e+02 |
| V09V10V11V12V17 | 27 | 10000 | 5.59e+69 | 7.68e+69 | 0.32 | 1.61e+02 |
| V09V10V11V13V17 | 27 | 10000 | 1.51e+70 | 9.90e+69 | -0.42 | 1.61e+02 |
| V09V10V12V13V17 | 26 | 10000 | 5.87e+67 | 4.74e+67 | -0.21 | 1.56e+02 |
| V09V11V12V13V17 | 27 | 10000 | 8.42e+69 | 1.19e+70 | 0.35 | 1.61e+02 |
| V10V11V12V13V17 | 26 | 10000 | 5.72e+66 | 4.14e+66 | -0.32 | 1.54e+02 |

Tabla G.12: Resultados modelo núcleo E.coli con cinco fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|--------------------|-----|-------|----------|----------|-----------------------|---------------------------|
| V08V09V10V11V12V13 | 28 | 10000 | 1.52e+70 | 2.44e+70 | 0.47 | 1.62e+02 |
| V08V09V10V11V12V17 | 28 | 10000 | 1.63e+73 | 4.46e+73 | 1.01 | 1.69e+02 |
| V08V09V10V11V13V17 | 28 | 10000 | 1.20e+73 | 1.39e+73 | 0.15 | 1.68e+02 |
| V08V09V10V12V13V17 | 27 | 10000 | 8.85e+70 | 9.19e+70 | 0.04 | 1.63e+02 |
| V08V09V11V12V13V17 | 28 | 10000 | 1.05e+73 | 1.46e+73 | 0.33 | 1.68e+02 |
| V08V10V11V12V13V17 | 27 | 10000 | 8.26e+69 | 8.17e+69 | -0.01 | 1.61e+02 |
| V09V10V11V12V13V17 | 28 | 10000 | 1.05e+73 | 1.50e+73 | 0.36 | 1.68e+02 |

Tabla G.13: Resultados modelo núcleo E.coli con seis fermentaciones prendidas.

| P | d | N | $CB \mu$ | $CG \mu$ | $\log(CG) - \log(CB)$ | $(\log(CG) + \log(CB))/2$ |
|-----------------------|-----|----------|----------|----------|-----------------------|---------------------------|
| V08V09V10V11V12V13V17 | 29 | 1.00e+04 | 1.24e+76 | 1.94e+76 | 0.45 | 1.75e+02 |

Tabla G.14: Resultados modelo núcleo E.coli con todas las fermentaciones prendidas.