Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Maestría en Ciencias
en Ciencias de la Computación

Sistema de almacenamiento de datos adaptativo y seguro en ambiente multi-nube

Tesis
para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Esteban Carlos López Falcón

Ensenada, Baja California, México
2018

Tesis defendida por

Esteban Carlos López Falcón

y aprobada por el siguiente Comité

Firma

Dr. Andrey Chernykh

Miembros del comité

Dr. Carlos Alberto Brizuela Rodríguez

Dr. Edgar Leonel Chávez González

Dr. Raúl Rivera Rodríguez

Dr. Jesús Favela Vara
Coordinador del Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez
Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Esteban Carlos López Falcón** como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación

Sistema de almacenamiento de datos adaptativo y seguro en ambiente multi-nube

Resumen aprobado por:

_____

Dr. Andrey Chernykh
Director de tesis

Los problemas de seguridad relacionados con cómputo en la nube, así como las soluciones propuestas en la literatura son uno de los temas importantes para la investigación en cómputo en la nube. Sin embargo, hay muchos problemas sin resolver relacionados con el almacenamiento en la nube. En esta tesis, implementamos un modelo adaptativo de almacenamiento de datos basado en Esquemas de Compartición de Secretos (SSS) y el Sistema Numérico de Residuo Redundante (RRNS). Propusimos seis estrategias para minimizar la pérdida de información y el tiempo de carga y descarga de datos en la nube. Evaluamos estas estrategias en siete proveedores de almacenamiento en la nube (CSP). Estudiamos la correlación de la configuración del sistema con la probabilidad de pérdida de información, velocidad de acceso a CSP y velocidades de codificación, decodificación. Presentamos evidencia experimental de que la estrategia AdaptiveSecurity que considera los CSP con las mejores velocidades de carga y luego, después del almacenamiento, migra los datos a los CSP con la menor probabilidad de pérdida de información muestra un mejor comportamiento de rendimiento en el análisis bi-objetivo y multi-objetivo.

Palabras clave: Pérdida de información, almacenamiento en la nube, Sistema Numérico de Residuo, seguridad, esquemas de compartición de secretos.

Abstract of the thesis presented by **Esteban Carlos López Falcón** as a partial requirement to obtain the Master of Science degree in Computer Science.

Adaptive and secure Multi-Cloud data storage system

Abstract approved by:

_____

Dr. Andrey Chernykh
Thesis Director

Security issues related to Cloud Computing as well as all solutions proposed in the literature are some of the high topics for research. However, there are unsolved problems regarded to cloud storage. In this thesis, we implement an adaptive model of data storage based on Secret Sharing Schemes (SSS) and Redundant Residue Number System (RRNS). We proposed six strategies to minimize information loss, time to data upload and download into the cloud. We evaluate these strategies on seven Cloud Storage Providers (CSPs). We study a correlation of system settings with the probability of information loss, speed of access to CSPs, and encoding/decoding speeds. We demonstrate that the AdaptiveSpeed strategy, which considers the CSPs with the best upload access speeds and then, after storing, migrates the data to the CSPs with the least probability of information loss shows better performance in our bi-objective and multi-objective analysis.

Keywords: Data loss, cloud storage, Residue Number System, security, Secret Sharing Schemes.

Dedicatoria

Agradecimientos

Tabla de contenido

Lista de figuras

Lista de tablas

Capítulo 1. Introducción

Gracias a la alta demanda de recursos computacionales en la nube, existe también una alta oferta de proveedores y no es necesario elegir solo a uno. Basta con realizar una búsqueda en internet acerca de servicios en la nube, para obtener un aproximado de un millón de resultados que varían entre artículos que explican qué es y cómo funciona la nube, listas de opinión sobre los mejores proveedores de servicios en la actualidad, etc. Sin embargo, cada lista de mejores proveedores de servicios puede variar, ya que la elección de proveedor, normalmente se realiza en base a preferencias personales.

En el caso de almacenamiento en la nube, es común encontrar que los proveedores ofrecen una cantidad de espacio de almacenamiento gratuito, por lo que si uno decide utilizar distintos proveedores, es posible conseguir una cantidad considerable de espacio de almacenamiento gratuito. Sin embargo, el manejo de distintas cuentas y recordar dónde se almacenó cierto archivo puede llegar a ser una tarea tediosa. Para facilitar el manejo de distintas cuentas de almacenamiento en la nube, existen aplicaciones como MultCloud (https://www.multcloud.com), Odrive (https://www.odrive.com), CloudFuze (https://www.cloudfuze.com), entre otros. A pesar de su utilidad, es necesario recordar en dónde se almacenó la información y realizar respaldos ocasionales.

Además, uno de los temas más preocupantes es el de seguridad de información. Los usuarios se encuentran temerosos al no tener su información de forma física en su ordenador, y con algo de razón ya que los proveedores de servicio son constantemente victimas de ciberataques (Morgan, 2018).

En este trabajo se plantean distintas estrategias para almacenar información utilizando un esquema de compartición de secretos basado en RRNS (Sección 2.2.6) para atacar el problema de seguridad de la información, distintos proveedores de almacenamiento en la nube para atacar el problema de disponibilidad, confiabilidad y evitar bloqueo por el proveedor. En las figuras 1 y 2 se muestra el flujo básico por el que pasa un secreto (archivo) para poder ser almacenado. Primero, el usuario elige el secreto que desea almacenar. Después, se le aplica algún algoritmo de compartición de secretos (Figura 1) y se divide el secreto en n fragmentos para ser almacenados en nubes distintas. Para poder recuperar el secreto, la cantidad de fragmentos que se tienen que utilizar, depende de la configuración que se eligió en el esquema de compartición de secretos. En este trabajo, una configuración se identifica por la tupla (k, n) dónde k representa la cantidad de fragmentos requeridos para recuperar la información y n la cantidad

de fragmentos en los que se dividió la información. El ejemplo ilustrado en las figuras 1 y 2 es una configuración $(2,3)$.

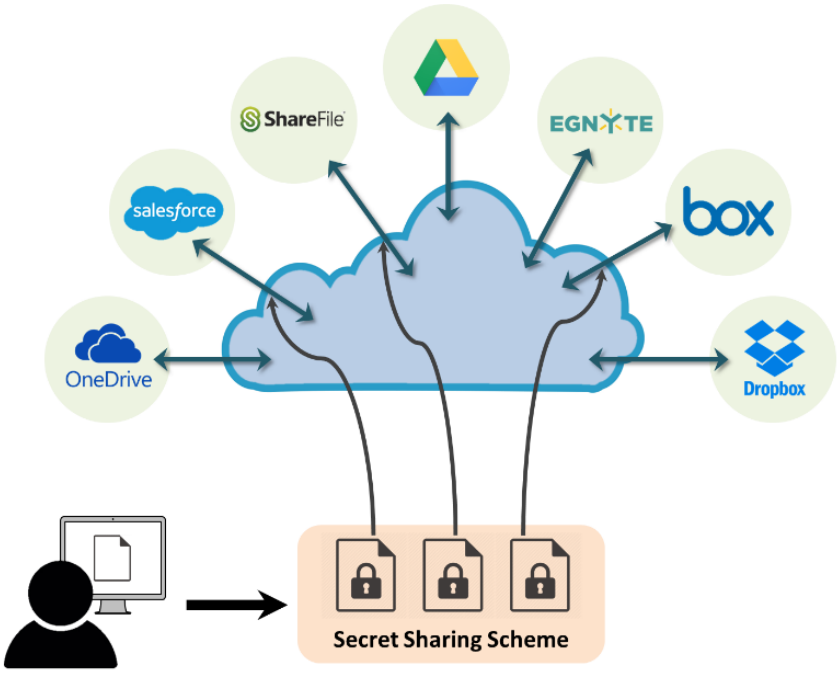

Figura 1. Carga de archivos encriptados a la multi-nube.

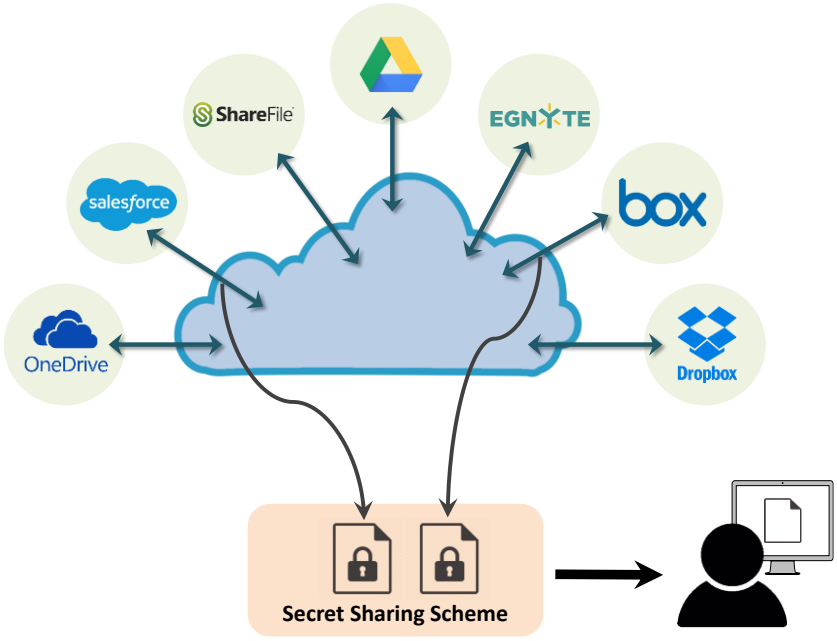

Figura 2. Descarga y decodificación de archivos de la multi-nube.

1.1 Antecedentes

En la actualidad, existen diversos algoritmos criptográficos que nos permiten modificar mensajes con el objetivo de ocultar su significado. Los dos trabajos más conocidos debido a su gran impacto son el publicado por Ron Rivest, Adi Shamir y Leonard Adleman en 1978 conocido como RSA (Rivest, Shamir, y Adleman, 1978), y el seleccionado por el Instituto de estándares y tecnologías (NIST) como el estándar avanzado de encriptación (AES) (Daemen y Rijmen, 2003), inicialmente publicado como Rijndael en 1998.

En 1979, Adi Shamir, mencionado en el párrafo anterior, contribuyó con la creación del algoritmo RSA, colocó el primer cimiento en los esquemas de umbral tipo (k, n) o esquemas de compartición de secretos, con su artículo *How to share a secret* (Shamir, 1979), al utilizar interpolación de polinomios como base para su sistema. El mismo año y de forma totalmente independiente George Blakley (Blakley, 1979) introdujo un sistema bastante similar al de Shamir, con la principal diferencia en que el trabajo de Blakley se basa en principios de la geometría finita y es menos eficiente en espacio que el sistema de Shamir.

Posteriormente en 1983, Maurice Mignotte (Mignotte, 1983) y los investigadores Charles Asmuth y Jhon Bloom (Asmuth y Bloom, 1983), publicaron casi en paralelo un esquema de umbral tipo (k, n) similar al de Adi Shamir, con la diferencia principal que en vez de utilizar interpolación de polinomios se basaron en aritmética modular y el teorema chino del residuo. Una de las diferencias entre el algoritmo de Asmuth-Bloom y Mignotte recae en que el primero es un esquema para compartir secretos perfectos y agrega ruido a la codificación y el segundo es estable computacionalmente y utiliza menos redundancia. Existe un compromiso entre un mejor rendimiento al utilizar Mignotte y una mejor seguridad al utilizar Asmuth-Bloom.

En 1989, Michal O. Rabin (Rabin, 1989) introdujo un algoritmo de dispersión de información (IDA) que a pesar de ser bastante similar a los sistemas de umbral tipo (k, n) y ser eficiente en espacio, carece de confidencialidad ya que $k - 1$ fragmentos pueden llegar a proporcionar información acerca del documento original.

Los esquemas de umbral tipo (k, n) son de especial interés para este trabajo ya que por su naturaleza distribuida son utilizados con frecuencia en el ambiente multi-nube. Tal es el caso de Ermakova, T. y Fabian, B. (2013) quienes plantean casos de estudio reales de hospitales europeos en arquitectura multi-nube y también evalúan el rendimiento en tiempo de los algoritmos de Shamir y Rabin. Sin embargo, no incluyen la implementación real de nubes de almacenamiento.

En un trabajo desarrollado por dos investigadores de la universidad de Mumbai (UMIT), Pundkar, S. y Shekokar, N. (2016), implementaron el algoritmo de Shamir con el esquema (2,3) utilizando las nubes de almacenamiento de Google Drive, Dropbox y OwnCloud (Tabla 2) para un portal estudiantil inter-campus en el que almacenan fotos y videos principalmente. A pesar de utilizar nubes reales, el hecho de contar con solo 3 proveedores de almacenamiento en la nube, limita las capacidades en seguridad del sistema.

En cómputo en la nube, se utilizan con frecuencia técnicas criptográficas para atacar los problemas de confidencialidad y privacidad. Como lo es el trabajo de Marium, S. et al. (2012), quienes propusieron una implementación del protocolo de autenticación extendido (EAP) junto con el protocolo de autenticación por *challenge-handshake* (CHAP) para atacar los problemas de autenticación y autorización, además de encriptar los mensajes con RSA. Al encriptar el mensaje con RSA y transferir con EAP-CHAP, se logra un alto nivel de seguridad en transferencia de datos.

El Sistema de Almacenamiento Dinámico y Seguro (DSSS) propuesto por Rathanam, G. et al. (2014), utiliza el algoritmo RSA para encriptar la información, técnica *Huffman* para comprimir antes de almacenar, y una vez almacenada la información realiza revisión remota de la integridad de la misma para detectar amenazas.

En el trabajo de Babitha, M. et al. (2016), utilizaron una base de datos alojada en GoDaddy (https://godaddy.com/), crearon una aplicación para almacenar información utilizando los lenguajes de programación Java y JSP. Para garantizar la seguridad de la información, decidieron utilizar la versión de 128 bits de AES para encriptar los datos antes de ser almacenados, además para que el usuario pueda compartir la información de forma segura, cada archivo almacenado cuenta con una contraseña elegida por el usuario. Lamentablemente, si se compromete la contraseña elegida por el usuario, la información puede ser comprometida.

En el ambiente multi-nube, existen trabajos que emplean el uso de algoritmos para compartir secretos para modelar sistemas de almacenamiento distribuido. En el trabajo de Chervyakov, N. et al (2017), se hace uso de un esquema al que ellos llaman AR-RRNS el cual combina propiedades de RRNS para dividir y repartir secretos y estrategias de aproximación numérica para reducir el costo computacional de las divisiones de números enteros grandes requeridas para recuperar el secreto original.

En el trabajo de Miranda-López V. et al. (2018), se realiza un análisis experimental con once proveedores de almacenamiento reales, implementan los algoritmos de Asmuth-Bloom y Maurice Mignotte y evalúan la diferencia en velocidades de carga y descarga al utilizar distintas configuraciones (k, n) además del tiempo de codificación y decodificación. Al utilizar un sistema de almacenamiento distribuido basado en un esquema para compartir secretos y múltiples nubes en lugar de una, se pueden atacar los problemas de pérdida de información, negación de acceso por un periodo de tiempo elevado, y fuga de información.

DEPSKY (Bessani et al., 2013) es un prototipo desarrollado en Java, enfocado a trabajar en el ambiente multi-nube. DEPSKY combina técnicas de encriptación, replicación y codificación. Distribuye los secretos y replica la información entre los proveedores. Siendo más específicos, DEPSKY Utiliza RSA con llaves de 1024 bits para encriptar las firmas de autenticación, SHA-1 para hash criptográfico, AES para encriptar la información, PVSS para para repartir la llave creada para la información a ser almacenada entre los distintos proveedores, y Reed-Solomon para verificación de errores. En general, el prototipo creado por DEPSKY ataca los problemas de disponibilidad, confiabilidad, bloqueo por proveedor y tolerancia a fallas. Sin embargo, el uso de DEPSKY representa un aumento en costo monetario considerable.

En el trabajo de Celesti, A. et al. (2016), se propone combinar las propiedades de RRNS con el algoritmo de encriptación AES 256-bits, de tal forma que cada fragmento generado al utilizar RRNS se encripta con AES 256-bits antes de ser enviado a un proveedor de almacenamiento en la nube, aumentando así la seguridad de cada fragmento. Con esto, ellos garantizan que a pesar de que todos los proveedores participantes en el esquema para compartir secretos se pusieran de acuerdo para intentar obtener información del documento original, tendrían que decodificar cada uno de los fragmentos almacenados primero, lo cual es prácticamente imposible. Una de las desventajas es el aumento en la cantidad de información a almacenar, sin embargo, en sus experimentaciones muestran que la introducción de AES no representa una sobre carga significativa en el sistema al utilizar archivos mayores a 100 MB. No obstante, no se muestra el análisis de cómo afecta el tiempo de codificación y decodificación de AES más la carga y descarga de los fragmentos. En la Tabla 1 se presenta una comparativa de los trabajos mencionados en esta sección.

Tabla 1. Comparación de características de trabajos en la literatura.

Autor	Multi-nube	SSS	Encriptación simétrica	Encriptación asimétrica	Análisis Teórico	Análisis Experimental	Seguridad	Interpolación	CRT	Almacenamiento de datos	Real CSPs	Privacidad
Rivest, R. et al. (1978)				●	●							
Rijmen, V. y Daemen, J. (2003)			●		●							
Sahmir, A. (1979)		●			●			●				
Blakley, G. (1979)		●			●							
Asmuth, C. y Bloom, J. (1983)		●			●				●			
Mignotte, M. (1983)		●			●				●			
Rabin, M. (1989)		●			●							
Ermakova, T. y Fabian, B. (2013)	●	●				●				●		
Pundkar, S. y Shekokar, N. (2016)	●	●				●		●		●		
Marium, S. et al. (2012)				●		●				●		
Rathanam, G. et al. (2014)				●		●				●		
Babitha, M. et al. (2016)			●			●	●			●		●
Miranda-López V. et al. (2018)	●	●				●	●		●	●	●	●
Bessani, A. et al. (2013)	●	●	●	●		●	●	●		●	●	●
Celesti, A. et al. (2016)	●	●	●			●	●		●	●	●	●
Chervyakov, N. et al (2017)		●			●		●		●	●		●

1.2 Objetivos

1.2.1 Objetivo general

Diseñar y analizar algoritmos para sistemas de almacenamiento de datos en la multi-nube, confiables y capaces de adaptarse a cambios en las velocidades de acceso y probabilidades de falla de las nubes,

minimicen el tiempo de carga (T_{up}), tiempo de descarga (T_{dow}), y minimicen la probabilidad de pérdida de información $\Pr(k, n)$.

1.2.2 Objetivos específicos

1. Desarrollar y analizar algoritmos de optimización mono-objetivo, que considere velocidades de nubes con los siguientes objetivos independientes: tiempo de carga (T_{up}), tiempo de descarga (T_{dow}), y probabilidad de pérdida de información ($\Pr(k, n)$).

2. Desarrollar y analizar algoritmos adaptativos que trabajan con velocidades variables de nubes.

   2.1. Desarrollar algoritmos adaptativos que minimicen el tiempo de carga T_{up}, tiempo de descarga T_{dow} y la suma de ambas

   2.2. Desarrollar un algoritmo adaptativo que minimicen el tiempo de carga T_{up} y tiempo de descarga $(T_{up} + T_{dow})$ y minimice la probabilidad de pérdida de información ($\Pr(k, n)$).

3. Algoritmos adaptativos de redistribución que trabajan con velocidades dinámicas de las nubes.

   3.1. Diseñar un algoritmo de redistribución de datos que se adapte a los cambios de parámetros.

4. Realizar un análisis experimental de los algoritmos adaptativos.

1.3 Organización de la tesis

En esta sección se detalla cómo se estructura este trabajo de tesis, describiendo de forma breve los aspectos principales de cada capítulo.

En el Capítulo 2, se detallan los conocimientos fundamentales para esta investigación, se profundiza en los temas de cómputo en la nube, seguridad informática y optimización multi-objetivo.

El Capítulo 3 contiene el planteamiento del problema, la notación matemática que se utilizará durante el resto del escrito y los criterios de optimización que se van a considerar en este trabajo. También, se definen las estrategias de selección de nube y los métodos que se van a utilizar para la evaluación de estas. Se describe el funcionamiento general del sistema de almacenamiento multi-nube y el proceso de codificación, almacenamiento, descarga y decodificación.

En el Capítulo 4 se describe la configuración experimental utilizada para la caracterización de las nubes y las simulaciones de variabilidad en el sistema. Se presentan los resultados de las estrategias y los criterios.

Por último, en el Capítulo 5 se presentan las conclusiones y discusiones de los resultados obtenidos, las contribuciones realizadas y las limitaciones encontradas y se discute el posible trabajo a futuro.

Capítulo 2. Marco Teórico

2.1 Computo en la nube

De acuerdo con lo mencionado por Kaufman et al. (2011), el término de cómputo en la nube, viene evolucionando desde los años sesenta cuando J.C.R. Licklidier introdujo el término “red computacional intergaláctica” en la agencia avanzada de proyectos de investigación (ARPA) cuya premisa era una interconexión global de datos y programas.

El Instituto Nacional de Estándares y Tecnología (NIST) define cómputo en la nube como: “Un modelo que permite acceso ubicuo, conveniente y bajo demanda a una red compartida de recursos computacionales configurables (ej., redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser entregados y distribuidos con un manejo mínimo o interacción con el proveedor de servicios.” (Mell y Grance, 2011).

En 2010, Armbrust M. publicó los 10 obstáculos y oportunidades de crecimiento para el cómputo en la nube. A mi conocimiento, al menos la mitad de esos obstáculos publicados han sido prácticamente solucionados. Esto debido a que el cómputo en la nube ha crecido rápidamente. Sin embargo, la resolución de problemas importantes permite el crecimiento de nuevas tendencias que proporcionan nuevos retos.

2.1.1 Sistemas de almacenamiento en la nube

Hoy en día existen un gran número de sistemas de almacenamiento en la nube, por lo que resulta prácticamente imposible listarlos todos. En la Tabla 2, se presentan algunos ejemplos de proveedores de servicio de almacenamiento en la nube, listados por orden alfabético:

Tabla 2. Ejemplos de proveedores de almacenamiento en la nube.

Nombre	URL completo
Alibaba Cloud	https://www.alibabacloud.com/
Amazon Drive	https://www.amazon.com/gp/drive/about/
Box	http://box.com/
certain safe	https://certainsafe.com/
Dropbox	http://dropbox.com/
Egnyte	https://egnyte.com/
Elephant drive	https://home.elephantdrive.com/
FlipDrive	https://flipdrive.com/
Google Drive	https://www.google.com/drive/
Hubspot	https://www.hubspot.com/
iCloud	https://www.icloud.com/
IDrive	https://www.idrive.com/
Jumpshare	https://jumpshare.com/
JungleDisk	https://www.jungledisk.com/
Justcloud	http://www.justcloud.com/
MediaFire	https://www.mediafire.com/
Mega	https://mega.nz/
one backup	https://mozy.com/
One Drive	https://onedrive.live.com/
ownCloud	https://owncloud.org/
pCloud	https://www.pcloud.com/
Rackspace	https://www.rackspace.com/cloud
Salesforce	https://www.salesforce.com/
Sharefile	https://www.sharefile.com/
spideroak	https://spideroak.com/one/
storegate	https://www.storegate.com/gl/
SugarSync	https://www2.sugarsync.com/
sync	https://www.sync.com/
Windows Azure	https://azure.microsoft.com/en-us/services/storage/
Yandex Disk	https://disk.yandex.com/

A continuación, se describen brevemente los sistemas utilizados en este trabajo:

Egnyte (https://egnyte.com/) fundado en 2007. Ofrece inicio de sesión con doble verificación y encriptación para toda la información en transmisión con AES de 256-bits. Para aumentar la disponibilidad de los archivos, utilizan replicación con redundancia independiente en almacenamiento RAID. Los servidores se encuentran en un centro de datos de tipo II SAS 70.

Dropbox (http://dropbox.com/) fundado en 2007. Ofrece inicio de sesión con doble verificación y encriptación en todo momento con AES de 256-bits.

OneDrive (https://onedrive.live.com/) creado por Microsoft en 2007, inicialmente conocido como Windows Live Folders y después como Windows Live SkyDrive. Cuenta con servicio de sincronización de archivos y almacenamiento. Ofrece 5 Gb de almacenamiento gratuito. Encripta los datos en transmisión utilizando SSL. Su versión para negocios ofrece encriptación por archivos, es decir, le asigna una llave de encriptación a cada archivo.

Google Drive (https://www.google.com/drive/) es el servicio de sincronización y almacenamiento de archivos creado por Google en 2012. Ofrece 15 GB de almacenamiento gratuito. Sin embargo, este almacenamiento es compartido entre las aplicaciones de Google Photos, Gmail, Google Keep, etc. Los archivos son encriptados con AES-256 bits. Para autenticar y verificar la validez de los archivos, utiliza SHA1-HMAC. Además, agrega datos al azar a cada mensaje para incrementar la seguridad en contra de ataques de tipo *cypher-text*.

Box (http://box.com/) fundado en 2005 como un sistema para negocios diseñado para compartir archivos en línea. Ofrece almacenamiento gratuito de 10 GB. A cada archivo compartido por Box se le puede asignar una contraseña y una fecha de expiración para que en caso de que la liga compartida sea comprometida, esto no sea suficiente para que se efectúe robo de información.

ShareFile (https://www.sharefile.com/) inició en noviembre de 2005. Tal como su nombre en inglés lo indica, es una compañía cuyo producto se enfoca en la colaboración, sincronización y compartición segura de archivos. Cuenta con eliminación remota, encriptación y bloqueado por fallas de contraseña como características de seguridad. Además, el administrador puede restringir las herramientas de terceros instaladas por sus usuarios que pueden leer sus archivos.

SalesForce (https://www.salesforce.com/) fundado en 1999. Es una compañía cuyo enfoque principal es ventas y mercadeo en la nube, pero también puede ofrecer hasta 1 TB de almacenamiento con costo.

2.1.2 Multi-nube

Gracias a la facilidad de acceso de información y al bajo costo de recursos computacionales, el cómputo en la nube creció en popularidad. Surgieron un gran número de servicios de almacenamiento en la nube con versiones gratuitas. El uso continuo de estos servicios empezó a generar dudas sobre la confidencialidad de la información proporcionada por los proveedores de servicio y la incertidumbre en el ambiente (Tchernykh et al., 2016).

Con la necesidad de mitigar las dudas de los usuarios y para atacar el problema en el cual los usuarios solo pueden elegir a un proveedor, investigadores en los laboratorios de IBM (Basescu, C. et al., 2011) publicaron un modelo en el cual conectaron múltiples nubes, algo que llamaron internube o nube de nubes.

En el modelo propuesto por IBM, cada proveedor de servicios mantiene un estado y posibles transiciones de estado que son activadas por las acciones de los clientes. Si un proveedor de servicios falla por problemas físicos, todas las acciones de dicho proveedor se deshabilitan por tiempo indeterminado. Este modelo y sus algoritmos publicados hicieron posible el surgimiento de la multi-nube, la cual puede ser definida como el uso de múltiples servicios computacionales de diferentes proveedores de manera transparente, es decir que la interacción simule el uso de una sola nube.

2.1.3 Problemas de seguridad en cómputo en la nube

Los proveedores de servicios en la nube ponen a disponibilidad del usuario una gran variedad de servicios, pero el abordado por este trabajo es el de almacenamiento en la nube. Cuando se almacena información sensible en la nube, se desea que se mantenga segura. Las instalaciones de almacenamiento en la nube se encuentran vigiladas en todo momento y los proveedores de servicio afirman que incluso ellos no pueden

saber dónde se almacena la información de algún usuario en específico. Sin embargo, estas declaraciones no son suficientes para garantizar la seguridad de la información y cesar las dudas de los usuarios.

La seguridad en cómputo en la nube debe ser garantizada. Una de las soluciones atractivas para investigadores ha sido el uso de técnicas de criptografía. Al usar algoritmos de encriptación en los archivos antes de que sean almacenados en la nube, se agrega una nueva capa de seguridad. Esta posible solución da pie al surgimiento de otros problemas, los cuales se discutirán a lo largo de este documento.

Un problema que se tiene que tomar en cuenta es el de restricción a un solo proveedor, esto ocurre cuando un consumidor es dependiente de algún producto o servicio de un proveedor sin poder utilizar el de otro proveedor obteniendo un trato similar. Si un usuario es dependiente del almacenamiento con un solo proveedor, su información se encuentra expuesta a los fallos de este.

2.2 Seguridad informática

El término seguridad informática abarca un gran número de áreas y actividades, desde vigilancia de hardware, protección de información, tolerancia a fallas entre otros. Una de las actividades en seguridad computacional es la de prevención a la interrupción de servicio. La interrupción de servicio, de acuerdo con Morrie Gasser (1988), se puede definir como una reducción temporal del rendimiento del sistema, ya sea por fallos del sistema al ser reiniciado o por algún fallo que genere la pérdida permanente de información. El enfoque de seguridad de este trabajo es el de interrupción de servicio, de tal manera que un sistema con menos interrupciones de servicio es más seguro.

2.2.1 Técnicas criptográficas

La criptografía es la ciencia de escribir de forma secreta con el objetivo de esconder el significado de algún mensaje. En cómputo en la nube la criptografía se utiliza para mantener la información distribuida de forma segura. Las técnicas de encriptación de información se pueden clasificar en dos subconjuntos: simétricas y asimétricas.

En un esquema simétrico o de llave privada, existen dos funciones prácticamente similares llamadas encriptación y decodificación, también una llave que se mantiene como secreto a los no participantes. La llave se utiliza para encriptar y decodificar el mensaje, si la llave se compromete, cualquier persona puede decodificar el mensaje. Debido a que no es fácil compartir la llave sin que esta se comprometa, esta técnica es más utilizada cuando no se requiere compartir dicha llave, esto es, cuando la persona que genera la información es la misma que la utiliza.

En un esquema asimétrico o de llave pública, al igual que en el esquema el simétrico, existe un método de encriptación, decodificación y una llave privada, pero también existe una llave pública. La llave pública se utiliza para encriptar el mensaje y la llave privada para decodificarlo (Paar & Pelzl, 2010).

El algoritmo simétrico más popular es el Estándar Avanzado de Encriptación (AES), está basado en permutaciones lineales y transformaciones. Existen tres diferentes versiones del algoritmo, AES-128, AES-192 y AES-256 donde los números representan el tamaño de la llave en bits (Daemen y Rijmen, 2003).

Por otro lado, el algoritmo asimétrico más popular es el Rivest, Samir y Adleman (RSA), el cual debido a su costo computacional, es mejor si es utilizado para encriptar información de pocos bits (Rivest et al., 1978).

2.2.2 Código de borrado

Los códigos de borrado o Erasure Codes (EC) son una técnica de recuperación y detección de errores, cambia un mensaje de k símbolos por un mensaje de n dónde $n > k$. La creación de mensajes se puede generalizar a una interpolación polinomial. Primero, se define un grupo finito F de tamaño al menos n. La entidad que envía el mensaje, enumera los símbolos que contienen desde 0 hasta $k - 1$, luego se realiza una interpolación de $p(x)$ de grado $k - 1$. Después, se construye un mensaje con $p(k), \dots, p(n - 1)$. Dicho mensaje se puede recuperar utilizando la misma interpolación si los primeros k llegaron a su destino exitosamente (Lin et al., 2014).

2.2.3 Código Reed-Solomon

Los códigos Reed-Solomon pueden detectar y corregir múltiples errores de símbolos. Al agregar t símbolos de verificación a los datos originales, Reed-Solomon puede detectar cualquier combinación de hasta t símbolos erróneos, o corregir hasta $\lfloor t \,/\, 2 \rfloor$ símbolos. También son adecuados para corregir errores de ráfagas múltiples de bits, ya que una secuencia de $b + 1$ errores de bits consecutivos puede afectar a lo más dos símbolos de tamaño b. La elección de t depende del diseñador del código. Las aplicaciones más comunes para estos códigos son CDs, DVDs, Blu-ray, WiMAX, código QR y sistemas de almacenamiento como RAID 6. También son utilizados en comunicación satelital (Reed y Solomon, 1960).

2.2.4 Replicación dinámica

La replicación de información consiste en colocar de forma estratégica copias de dicha información para incrementar su disponibilidad, confiabilidad y tolerancia a fallas. La replicación dinámica es uno de los grupos en los cuales se pueden clasificar las estrategias de replicación. Las estrategias de replicación dinámica consisten en responder a cambios de parámetros dentro del sistema eligiendo qué, cuándo y dónde se va a replicar. Esta estrategia es mejor utilizada en ambientes en los cuales los nodos donde se almacena la información pueden entrar y salir del sistema a placer (Tos et al., 2015).

2.2.5 Shamir's Secret Sharing Scheme

Un esquema de compartición de secretos o *Secret Sharing Scheme (SSS)*. Es un esquema de seguridad donde dada una llave x un administrador la divide en $y_1, \dots, y_n$ fragmentos llamados secretos, de tal manera que la llave puede ser reconstruida utilizando cualesquiera r secretos, además no se puede obtener ningún fragmento de información utilizando $r - 1$ o menos secretos, después de realizar la división, las sombras son distribuidas a los distintos participantes. La esencia de este esquema lo hace popular en sistemas distribuidos tales como el cómputo en la nube.

El primer esquema de este tipo fue diseñado por Adi Shamir y se basa en la propiedad de los polinomios que dice que dados k puntos en un plano de dos dimensiones $(x_1, y_1), \dots, (x_k, y_k)$ con distintas

x_i, existe un y solo un polinomio $q(x)$ de grado $k-1$ tal que $q(x_i) = y\ \forall i$. Por lo tanto, a la hora de dividir la información D en secretos, se elige un polinomio de grado $k-1$ al azar $q(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$ tal que $a_0 = D$ y se evalúan $D_1 = q(1), \ldots, D_i = q(i), \ldots, D_n = q(n)$. Cualquier subconjunto de tamaño k, de los D's evaluados, junto con sus índices, permite encontrar los coeficientes de $q(x)$ mediante interpolación de polinomios y después evaluar $D = q(0)$ (Shamir, 1979).

2.2.6 Sistema numérico de residuo

El sistema numérico de residuo (RNS) es un paradigma de la teoría de números ampliamente conocido. Se utiliza para representar un número entero muy grande mediante el uso de números enteros más pequeños, esto permite la reducción en complejidad de ciertas operaciones computacionales. Se basa principalmente en el teorema chino del residuo descubierto por Sunzi Suajing en el siglo III (Flores, 1969).

Una descripción general del sistema dada por Ferruccio Barsi et al. (1973) es la siguiente: "Dado un conjunto de n números enteros positivos primos relativos por pares $m_1, m_2, \ldots, m_n$ llamados módulos, los enteros no negativos X en el intervalo $[0, M)$, dónde $M = m_1 \cdot m_2 \cdot \ldots \cdot m_n$, son únicamente representados por n-tuplas $x_1, x_2, \ldots, x_n$ de sus residuos modulo m_i $(x_i = |X|_{m_i}, i = 1,2, \ldots, n)$." El entero X en el intervalo $[0, M)$ se puede obtener a partir de sus dígitos $x_i's$ con la fórmula $\left(\sum_{i=1}^{k} x_i \frac{M}{m_i} B_i\right) mod\ M$ donde B_i es el número tal que $B_i \frac{M}{m_i} + b_i m_i = 1$.

El sistema numérico de residuo redundante (RRNS) utiliza un conjunto de $(n+r)$-tuplas para representar un entero del intervalo $[0, M)$, pero usa $m_1, m_2, \ldots m_n, m_{n+1}, \ldots, m_r$ módulos y $x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_r$ dígitos, los $m_{n+1} \ldots m_r$ y $x_{n+1}, \ldots, x_r$ son llamados módulos redundantes y dígitos redundantes respectivamente. Se puede encontrar un ejemplo en la Sección 3.4.1.

Uno de los usos comunes del sistema numérico de residuo redundante es el almacenamiento de datos. En este caso, la seguridad depende de los parámetros de k y n que se vayan a utilizar, la selección apropiada de estos parámetros provee el nivel de seguridad computacional necesario. Por otro lado, las propiedades del sistema permiten la corrección y detección de errores de fallas que pudieran ocurrir. Chervyakov, et al. (2017) muestra que la confiabilidad del sistema depende de $r = n - k$. Mientras mayor sea el valor de r, más confiable es el sistema, aunque con mayor redundancia de datos.

2.2.7 Maurice Mignotte Secret Sharing Scheme

En el esquema propuesto por Maurice Mignotte, el secreto es un número entero. En este esquema, se requieren $d_1, \dots, d_n$ primos relativos por pares, el secreto es un entero $S, a \leq S \leq b$, dónde a y b son enteros dados y $0 < a < b$. Para obtener el esquema (k, n) se toman $d_1, \dots, d_n$ tales que el producto de k de los d_j es mayor a b y el producto de $k-1$ de los d_j es menor a a. La información se divide de la forma $x_j = S \; mod \; d_j, \;\; 1 \leq j \leq n$ y se recupera con el teorema chino del residuo (Mignotte, 1983).

2.2.8 Asmuth-Bloom Secret Sharing Scheme

Al igual que el esquema propuesto por Mignotte, Asmuth-Bloom es un esquema de compartición de secretos basado en el teorema chino del residuo y se define a continuación. Sean k y n dos números enteros positivos tales que $0 < k + 1 \leq n$ y dada una secuencia de números primos relativos $m_0, m_1, \dots, m_n$ que cumplen con las siguientes propiedades:

$$m_0 < m_1 < \cdots < m_n \tag{1}$$

$$\prod_{i=1}^{k+1} m_i > m_0 \prod_{i=0}^{k-1} m_{n-i} \tag{2}$$

Dado un secreto s y un número generado al azar r de tal forma que $s' = s + rm_0 < \prod_{i=1}^{k+1} m_i$. Se comparte s al hacer $s_i = s' \; mod \; m_i \; \forall \; 1 \leq i \leq n$. Para reconstruir el secreto cualquier subconjunto de A participantes tales que $|A| \geq k + 1$ es suficiente, primero se utiliza el teorema chino del residuo para obtener $x \equiv s_i \; mod \; m_i, \forall i \in A$ y después se reduce $x \; mod \; m_0$ para obtener el secreto original (Asmuth y Bloom, 1983).

2.3 Optimización multi-objetivo

La optimización matemática estudia el problema de encontrar el mejor elemento dentro de un conjunto de alternativas factibles de acuerdo a algún criterio o función objetivo. Se escribe de la forma

$$optimizar\ f(x)$$

$$sujeto\ a\ x \in X,$$

dónde X es el conjunto de las soluciones factibles.

Un problema de optimización de múltiples objetivos se escribe de la forma

$$optimizar\ F(x) \coloneqq (f_1(x), f_2(x), \dots, f_n(x))$$

$$sujeto\ a\ x \in X,$$

dónde $f_1, f_2, \dots, f_n$ son las funciones objetivo en X y optimizar significa encontrar un elemento $x' \in X$ tal que ningún valor $F(x), x \in X$ es menor que el valor de $F(x')$ si el problema es de minimización o mayor si es de maximización, durante el resto de este texto se utiliza el término para problemas de minimización. El elemento x' no necesariamente es el mejor, solamente no es peor que ninguna otra solución. Esto quiere decir que existe un conjunto de soluciones optímales al problema. Por lo tanto, resolver un problema multi-objetivo se refiere a encontrar el conjunto de soluciones optímales. Los problemas de optimización bi-objetivo son un subconjunto de los problemas de optimización multi-objetivo con solo dos funciones objetivo.

Para definir cuando un elemento x' es mejor que otro, se utiliza la relación de dominancia. Existen dos relaciones de dominancia, dominancia débil y dominancia fuerte. Se dice que un elemento x' domina fuertemente a un elemento x, si el valor de cada función objetivo $f_j\ \forall j \in \{1,2, \dots, n\}$ de x'es menor que los de x. Se dice que un elemento x' domina débilmente a un elemento x, si el valor de al menos una función objetivo $f_j\ \forall j \in \{1,2, \dots, n\}$ de x'es estrictamente menor que los de x, y además el resto de los valores de las funciones de x' son menores o iguales a los de x. Un elemento es óptimo de Pareto si no es dominado fuertemente por algún otro elemento del conjunto de soluciones factibles. Al conjunto de todas

las soluciones óptimas de Pareto se le denomina conjunto de Pareto. El mapeo del conjunto de Pareto al espacio objetivo se le conoce como frente de Pareto (Figura 3) (Luc, 2016).

Cualquier solución que no pertenece al frente puede considerarse de calidad inferior comparado con las que sí se encuentran en este. La selección entre las soluciones incluidas en el frente de Pareto depende de la preferencia del sistema. Si se considera que un objetivo es más importante que el otro, entonces se da preferencia a aquellas soluciones que son casi óptimas en el objetivo preferido, incluso si los valores del objetivo secundario no están entre los mejores obtenidos.

En general, la optimización multi-objetivo suele encontrar un conjunto de soluciones conocido como conjunto óptimo de Pareto (Deb y Kalyanmoy, 2001). El objetivo es elegir la posible solución más adecuada y obtener un conjunto de soluciones no dominadas que resulten en una buena aproximación al frente de Pareto-óptimo. Dos características importantes de una buena técnica multi-objetivo son la convergencia al frente de Pareto y la diversidad, para obtener un muestreo del frente lo más completo posible.

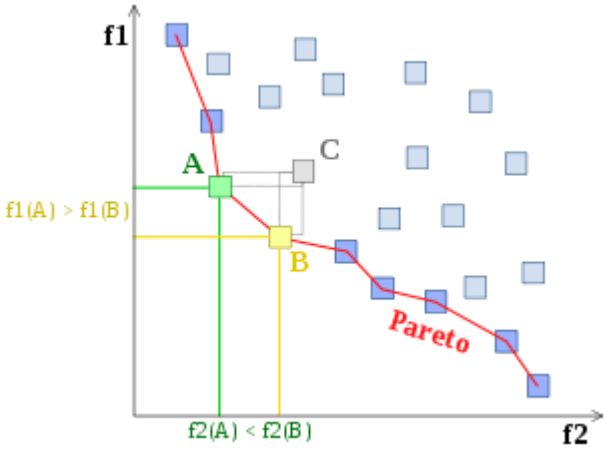

Figura 3. Frente de Pareto-óptimo

Capítulo 3. Metodología

3.1 Planteamiento del problema

En su forma más básica, el problema de almacenamiento consiste en la necesidad de guardar un objeto. Este objeto, puede tener distintas formas, joyas, dinero, información, etc. El problema de almacenamiento de información consiste en guardar de alguna forma información digital para ser consultada en algún momento. Para almacenar información existen distintos métodos, se puede grabar dicha información en algún dispositivo de almacenamiento (Discos duros, disquetes, CD, DVD, USB flash, etc.), etiquetarlo y colocarlo en algún lugar que pueda ser recordado.

La desventaja de almacenar información con el método anterior, es que si en algún momento se llega a perder o dañar el dispositivo de almacenamiento, la información se pierde. Para resolver este problema, se puede almacenar la información en varios dispositivos de almacenamiento, etiquetarlos y guardarlos en lugares distintos o en donde mismo, la desventaja de esto llega al momento que alguien desea robar la información, por lo que ahora tiene varios objetivos que puede robar. Para hacer que un ladrón no pueda obtener el mensaje que está grabado, se pueden utilizar distintas técnicas de codificación o difusión de información.

Las técnicas de dispersión de información (IDA) dividen la información I de tamaño $L = |I|$ en n piezas distintas de longitud L/m, de tal forma que m piezas sean suficientes para construir F, con esto el robo de una pieza no es suficiente para obtener toda la información, pero podría ser que esa pieza contenga información valiosa. Además, no hay un mecanismo de decodificación, por lo que con el robo de m piezas se puede obtener la información original sin tener que realizar trabajo extra. Generalmente, estos algoritmos mantienen una relación de $\frac{n}{m}$ cercana a uno para que sean eficientes en espacio.

Para tener mejor recuperación de errores, existen los códigos de tipo *erasure* cómo lo es Reed-Solomon. Este tipo de códigos, almacenan el mensaje original agregando espacio extra para detección y corrección de errores. Estos códigos son altamente utilizados en esquemas de discos con alta tolerancia a fallas.

En lugar de realizar el almacenamiento de información en dispositivos de almacenamiento físicos, se puede realizar en nubes de almacenamiento. Los problemas de seguridad en cómputo en la nube son

factores importantes para el almacenamiento y procesamiento de datos. Además de los problemas de seguridad y confiabilidad existentes en el cómputo distribuido tradicional, existen nuevos problemas de seguridad y confiabilidad. Estos, incluyen ataques a una máquina virtual, ataques a las claves de sincronización, etc. De acuerdo con la evaluación de expertos internacionales en el campo de la seguridad en la nube, existen riesgos de colusión de nubes en condiciones inciertas.

Para mitigar este tipo de incertidumbre y reducir los daños que puede causar, Tchernykh et al. (2018) proponen el algoritmo AC-RRNS basado en una modificación de los esquemas de compartición de secretos Asmuth-Bloom y Mignotte, el cual satisface la definición formal de seguridad computacional. En su trabajo, demuestran que la probabilidad de obtener la llave secreta para una configuración (k, n) probando todas las combinaciones posibles es menor a $(\frac{1}{2^{l \cdot k}})$, dónde 2^l es el número tal que $2^{l-1} < llaves < 2^l$ y k es el parámetro de la configuración (k, n).

Como menciona AlZain et al. (2011): “Garantizar la seguridad del cómputo en la nube es un factor de gran importancia, debido a que los usuarios, con regularidad, almacenan información de importancia con sus respectivos proveedores, se tiende a desconfiar de dichos proveedores.”. Esta desconfianza hacía los proveedores de servicio es una de las cuestiones del cómputo en una sola nube que permitió que el ambiente multi-nube creciera en popularidad.

Los sistemas desarrollados para hacer uso de múltiples nubes, pueden llegar a sufrir problemas como pérdida o escape de información y negación de acceso por un largo periodo de tiempo (Tchernykh et al., 2016).

Para atacar algunos de los problemas mencionados, investigadores han propuesto el uso de técnicas de encriptación, replicación y cambios de símbolos. Debido a la naturaleza distribuida de los esquemas de umbral tipo (k, n), se han vuelto una opción interesante a utilizar en sistemas multi-nube para investigación (Chervyakov et al., 2017), (Celesti et al., 2016).

Sin embargo, al alcance de mi conocimiento, no se ha propuesto aún un sistema de almacenamiento que adapte su esquema de seguridad y confiabilidad con respecto a los parámetros cambiantes presentes en el ambiente multi-nube, por ejemplo, tiempos de carga, de descarga, disponibilidad de las nubes, etc. Específicamente, dado un sistema de almacenamiento en un ambiente multi-nube que utilice un sistema de compartición de secretos con configuración (k, n), dónde los valores elegidos para k y para n impactan directamente la probabilidad de pérdida de información, tiempo de

carga y descarga, debería de adaptarse a los parámetros cambiantes y seleccionar las mejores nubes para almacenar la información, además de tomar en cuenta la configuración (k, n) a utilizar.

En este trabajo, para resolver el problema de almacenamiento de información, se utiliza un esquema de compartición de secretos basado en el teorema chino del residuo en multi-nube. Este esquema permite división de información con cierto nivel de redundancia, por lo que la perdida de alguna división no representa la pérdida total de la información, detección y corrección de errores, lo que permite que un cierto nivel de daño en un archivo no perjudique la información original, además de la posibilidad de realizar operaciones como la suma y multiplicación modular en los fragmentos divididos. Su mayor desventaja se encuentra en la complejidad computacional requerida para realizar divisiones de números grandes.

3.2 Definición formal

Considere un conjunto de m nubes $C = \{c_1, c_2, \dots, c_m\}$. Cada nube $c_j = \{u_j, d_j, err_j\}$ está caracterizada por la velocidad de carga u_j, velocidad de descarga d_j, probabilidad de falla err_j $\forall j = \{1, \dots, m\}$.

Un sistema de compartición de secretos basado en RRNS con una configuración (k, n), donde la información D se divide en n fragmentos para ser almacenados. Cada fragmento $i = \{s_i\}$ tiene un tamaño s_i $\forall i = \{1, \dots, n\}$.

Permítanos utilizar la siguiente notación:

D	Tamaño original de la información,
D_E	Tamaño de la información encriptada,
s_i	Tamaño del $i - ésimo$ fragmento original,
s_{Ei}	Tamaño del $i - ésimo$ fragmento encriptado,
u_j	Velocidad de carga de la $j - ésima$ nube.
d_j	Velocidad de descarga de la $j - ésima$ nube,
T_D	Tiempo de decodificación total,
T_E	Tiempo de encriptación total,
T_{up}	Tiempo de carga de la información encriptada,

T_{dow} Tiempo de descarga de la información encriptada,

V_s Velocidad de codificación y carga de información,

V_{ex} Velocidad de decodificación y descarga de información,

R Redundancia,

m_i $i-$ ésimo modulo,

$P_r(k,n)$ Probabilidad de pérdida de información en configuración (k,n).

err_j Probabilidad de falla en la $j-$ ésima nube,

La *Velocidad de almacenamiento* (V_s) representa qué tan rápido divide la información, se encripta y se almacena cada uno de los fragmentos en su respectiva nube. Se calcula como el tamaño original de la información D entre la suma del tiempo de encriptación T_E y el tiempo de carga $T_{up} = \sum_{i=1}^{n} \frac{s_{Ei}}{u_i}$:

$$V_s = \frac{D}{T_E + T_{up}} \tag{3}$$

La *Velocidad de extracción* (V_{ex}), representa qué tan rápido descarga la información de cada una de las nubes y se decodifica para que pueda ser usada, se calcula como el tamaño original de la información D entre la suma del tiempo de decodificación T_D y el tiempo de descarga $T_{dow} = \sum_{i=1}^{k} \frac{\mathrm{s_{Ei}}}{d_i}$

$$V_{\mathrm{ex}} = \frac{D}{T_D + T_{dow}}, \tag{4}$$

Suponga que cada uno de los fragmentos es descargado de forma secuencial de las nubes. El proceso de descarga termina cuándo k fragmentos son descargados de forma correcta. Existen dos escenarios. En el mejor de los casos, solo se necesitan los primeros k fragmentos de información para finalizar la descarga. El peor de los casos, ocurre cuando el sistema tiene que leer las k nubes más lentas, indexadas de $n-k+1$ a n.

La *Probabilidad de pérdida de información* ($\mathrm{P_r(k,n)}$) se calcula como:

$$P_r(k,n) = \sum_{A \in F_{n-k+1}}^{n} \prod_{j \in A} err_j \prod_{j^c \in A^c} (1 - err_{j^c}), \quad (5)$$

dónde el conjunto F_{n-k+1} es el conjunto de todos los posibles $n-k+1$ subconjuntos de C, y A^c es el complemento de los subconjuntos A y C.

3.2.1 Análisis de criterios

Redundancia (R) es la razón del tamaño original de la información D y el de la información codificada almacenada D_E. La cantidad de redundancia que manejan los esquemas de compartición de secretos, varía dependiendo del esquema a utilizar (Shamir's, Asmuth-Bloom, Mignotte, etc.). La Figura 4, muestra la redundancia presente en el esquema de Mignotte al utilizar un archivo de entrada de 100 MB.

La Figura 5, muestra la redundancia al utilizar el esquema de Asmuth-Bloom con un archivo de entrada de 100 MB, esta es mayor que la redundancia al utilizar Mignotte.

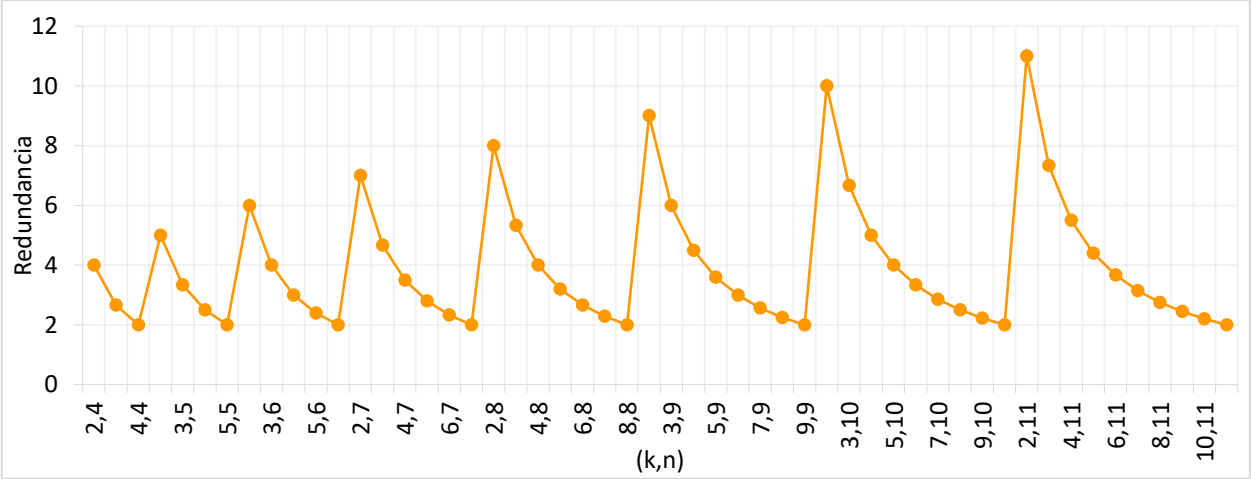

Figura 4. Redundancia con el esquema de Mignotte.

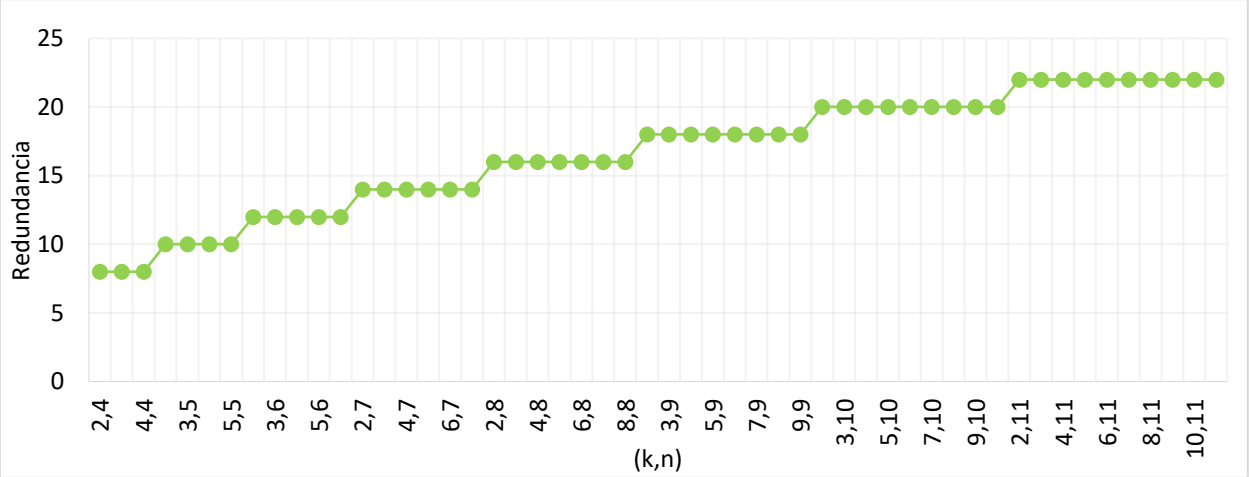

Figura 5. Redundancia con el esquema de Asmuth-Bloom.

Debido a que la velocidad de almacenamiento (V_s) depende del tiempo de encriptación, esta va a variar junto con el esquema de compartición de secretos que se esté utilizando. Existen esquemas de compartición de secretos cuyo tiempo de codificación es mayor a otros, esto debido a que algunos introducen ruido u otras estrategias para incrementar la seguridad del esquema, al hacer esto, realizan más operaciones y generalmente esto lleva un incremento de tiempo. En la Figura 6, se muestra la velocidad de almacenamiento de los esquemas Asmuth-Bloom y Mignotte al utilizar un archivo de entrada de 100 MB. Asmuth-Bloom introduce ruido para incrementar la seguridad del esquema, esto incrementa la redundancia de los fragmentos y hace que su velocidad sea menor. Para Mignotte, las configuraciones con menor redundancia (Figura 4) presentan mayor velocidad.

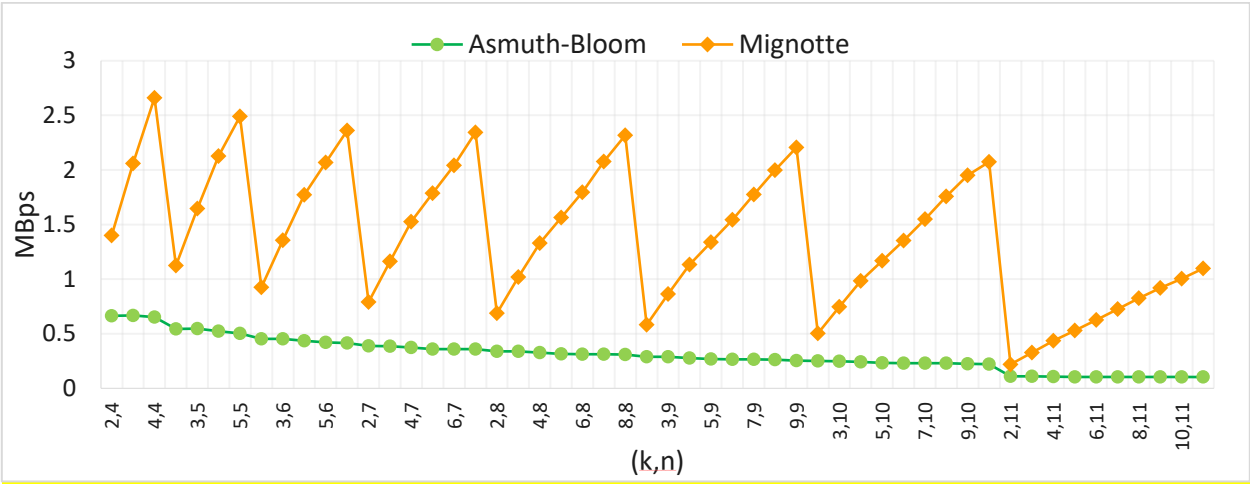

Figura 6. Velocidad de almacenamiento de Asmuth-Bloom y Mignotte (Miranda-López et al., 2018).

La velocidad de extracción (V_{ex}) se comporta de forma similar a la velocidad de almacenamiento, en el sentido que esta depende del tiempo de decodificación, y la decodificación va a variar junto con el esquema de compartición de secretos que se esté utilizando. En la Figura 7, se muestra la velocidad de extracción de los esquemas Asmuth-Bloom y Mignotte al utilizar un archivo de entrada de 100 MB. De nuevo, al igual que la velocidad de almacenamiento, el esquema de Asmuth-Bloom tiene menor velocidad que Mignotte, debido a que para eliminar el ruido introducido requiere de más operaciones, además que sus fragmentos manejan más redundancia que Mignotte (Figura 4 y Figura 5) y esto hace que el tiempo de descarga aumente, disminuyendo la velocidad. Para Mignotte, las configuraciones con menor redundancia (Figura 4) presentan mayor velocidad.

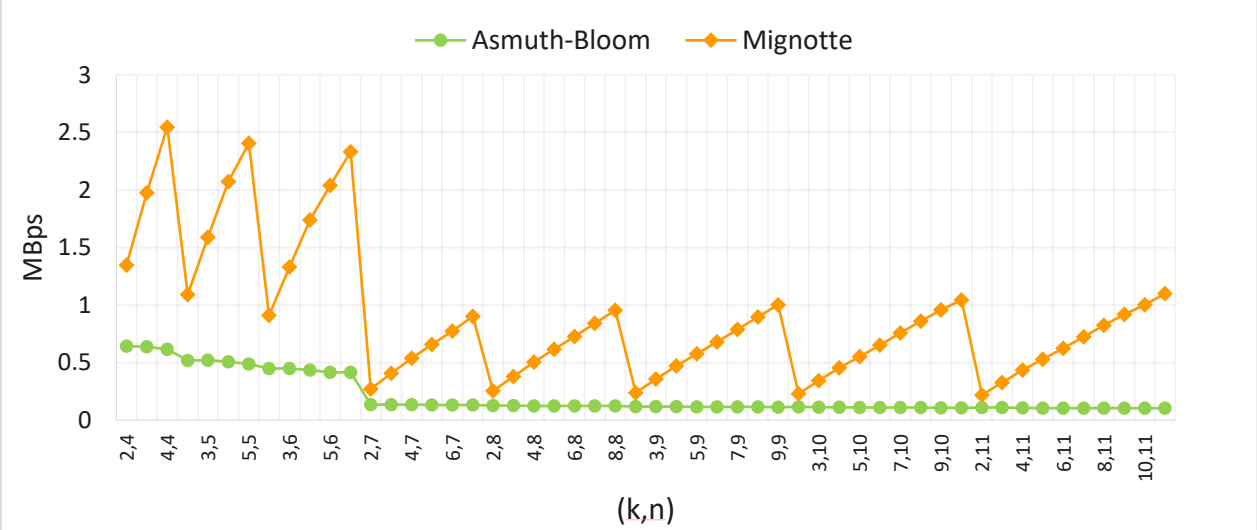

Figura 7. Velocidad de extracción de Asmuth-Bloom y Mignotte (Miranda-López et al., 2018).

En el trabajo de Chervyakov et al. (2017) calculan la probabilidad de pérdida de información cómo $P_r(k,n) = \sum_{j=n-k+1}^{n} C_n^j (err_j + DDoS_j)$, donde $err_j = 0.01^j \cdot 0.99^{n-j}$ y $DDoS_j = 0.05^j \cdot 0.95^{n-j}$, el comportamiento de esta función presenta un incremento en la probabilidad de falla cuando los parámetros de n y k se acercan (Figura 8). En este trabajo, se espera que la función de probabilidad presentada en la fórmula (5), tenga el mismo comportamiento.

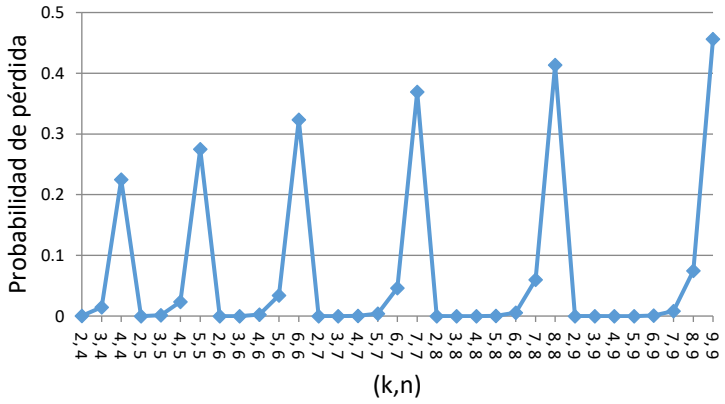

Figura 8. Probabilidad de pérdida de información contra la configuración (k, n).

3.3 Estrategias para selección de nubes

A continuación, se describen las estrategias propuestas en este trabajo para seleccionar los proveedores de almacenamiento en la nube que van a ser utilizados para alojar los fragmentos generados del archivo original por medio de un esquema de compartición de secretos.

Random. Esta estrategia selecciona n nubes disponibles de forma arbitraria.

BestUpload. Esta estrategia selecciona las n nubes disponibles con la mejor velocidad de carga.

BestDownload. Esta estrategia selecciona las n nubes disponibles con la mejor velocidad de descarga.

BestSecurity. Esta estrategia selecciona las n nubes disponibles con la menor probabilidad de falla.

AdaptiveSpeed. Sea $S_{u(n)}$ el conjunto de n nubes disponibles con la mejor velocidad de carga y sea $S_{d(n)}$ el conjunto de nubes con la mejor velocidad de descarga. Esta estrategia selecciona primero las nubes que pertenecen a $S_{u(n)}$ y ya que los fragmentos se encuentran almacenados, transfiere los fragmentos que se encuentran en las nubes $c \in (S_{u(n)} \backslash S_{d(n)})$ a las nubes $c \in (S_{d(n)} \backslash S_{u(n)})$.

AdaptiveSecurity. Sea $S_{u(n)}$ el conjunto de n nubes disponibles con la mejor velocidad de carga y sea $S_{pr(n)}$ el conjunto de nubes con la probabilidad de falla mínima. Esta estrategia selecciona primero las nubes que pertenecen a $S_{u(n)}$ y ya que los fragmentos se encuentran almacenados, transfiere los fragmentos que se encuentran en las nubes $c \in (S_{u(n)} \setminus S_{pr(n)})$ a las nubes $c \in (S_{pr(n)} \setminus S_{u(n)})$.

A continuación, en la Tabla 3 se clasifican las estrategias en base al conocimiento requerido para utilizarlas.

Tabla 3. Estrategias por conocimiento requerido.

Conocimiento	**Estrategia**
Sin conocimiento	*Random*
Velocidad conocida	*BestUpload* *BestDownload* *AdaptiveSpeed*
Probabilidad de falla conocida	*BestSecurity*
Velocidades y probabilidad de falla conocida	*AdaptiveSecurity*

3.4 Descripción del sistema

En esta sección, se describe de forma detallada el proceso de almacenamiento y de extracción de un archivo utilizando el sistema propuesto en este trabajo. El cual es una implementación del esquema de compartición de secretos AR-RRNS, excluyendo la aproximación numérica para facilitar su explicación.

Primero, es necesario seleccionar el conjunto de $n + r$ números primos relativos por pares tales que $m_0 < m_1 < \cdots < m_n < m_{n+r}$.

Existen diversos algoritmos para realizar esta tarea, el algoritmo utilizado en este trabajo es el Algoritmo 1. El algoritmo recibe de parámetros dos variables, n representa el tamaño del conjunto y l representa el exponente con base 2, que denota el número cuya suma con 1, es el número mínimo permitido en el conjunto. El Algoritmo 1, verifica si $2^l + 2$ es primo relativo de $2^l + 1$, para esto utilizamos la función $gcd(a, b)$ que puede ser cualquier algoritmo para encontrar el máximo común divisor de dos números enteros, por ejemplo, el algoritmo de Euclides, en caso de serlo, lo agrega al conjunto de primos relativos por pares, en caso de no serlo, no se realiza ninguna operación. Se continua con $2^l + 3$ realizando la misma verificación pero ahora con $2^l + 1$ y $2^l + 2$, esto se realiza de forma consecutiva hasta obtener

n números dentro del conjunto. En general, se realiza una búsqueda iterativa sobre todos los números enteros mayores a $2^l + 1$, hasta encontrar n números primos relativos por pares. Para esto, se utiliza la función $greatest\ common\ divisor$ (gcd), la cual se puede realizar con el Algoritmo 2, sobre cada número primo relativo previamente encontrado y el número entero correspondiente a la iteración.

Algoritmo 1: Get co-prime Set

```
1     procedure CoPrimeSet(n, l)
2     let m[1 … n] be a new array
3     m[1] = 2^l+1
4     flag = 1
5     for i = 2 to n do
6           m[i] = m[i − 1] + 1
7           flag = 2
8           While flag > 1 do
9                 flag = 1
10                for j = 1 to i − 1 do
11                      flag = flag*gcd(m[i], m[j])
12                end for
13                if flag > 1 then
14                      m[i] = m[i] + 1
15                end if
16          end while
17    end for
18    return m
19    end procedure
```

Después se calcula el rango dinámico del sistema, el cuál es $M = m_1 \cdot m_2 \cdot \ldots \cdot m_n$. También, llamamos $LenBlock(M)$ a la cantidad de bytes requeridos para representar M. Luego, ya que se tiene el archivo que se desea almacenar, se recorre el archivo derecha a izquierda en intervalos de tamaño $LenBlock(M)$, definimos el contenido de este intervalo como s_j para $j = \{1, \ldots, \lceil \frac{D}{LenBlock(M)} \rceil\}$, si el tamaño del archivo no es divisible por $LenBlock(M)$, se agregan ceros a la izquierda del último intervalo

leído. Se generan los fragmentos a almacenar de tal forma que el fragmento 1 contiene cada $s_j \ mod \ m_1$, el fragmento 2 cada $s_j \ mod \ m_2$ y así sucesivamente (Figura 9).

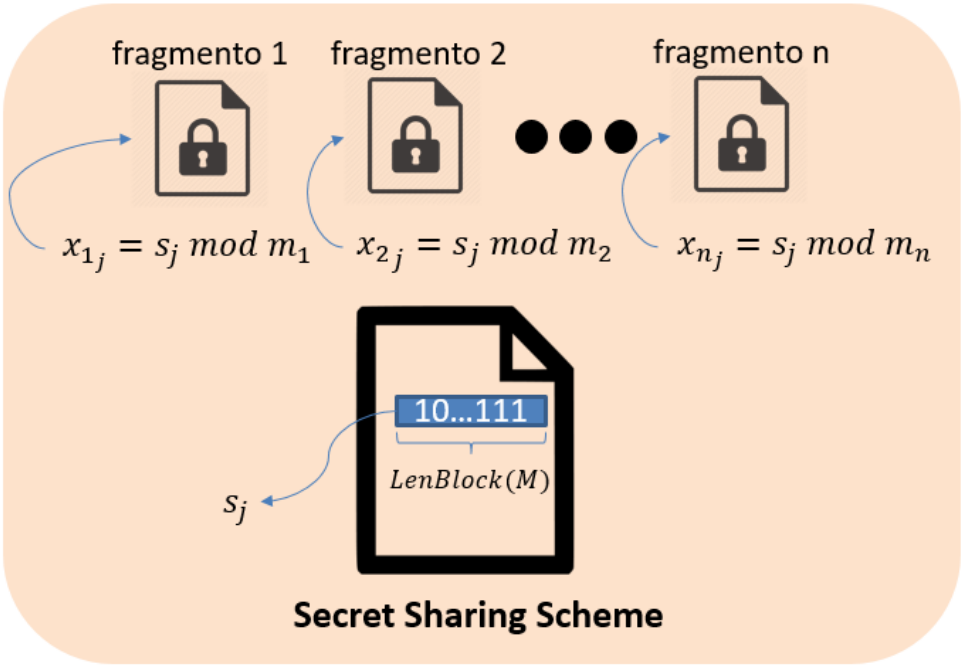

Figura 9. Elaboración de los fragmentos a partir del archivo original.

A continuación, en base a una estrategia, se seleccionan las nubes en las cuales se van a almacenar los fragmentos generados (Figura 10). En el caso de que se use una estrategia adaptativa, algunos fragmentos pueden moverse de una nube a otra (Figura 11).

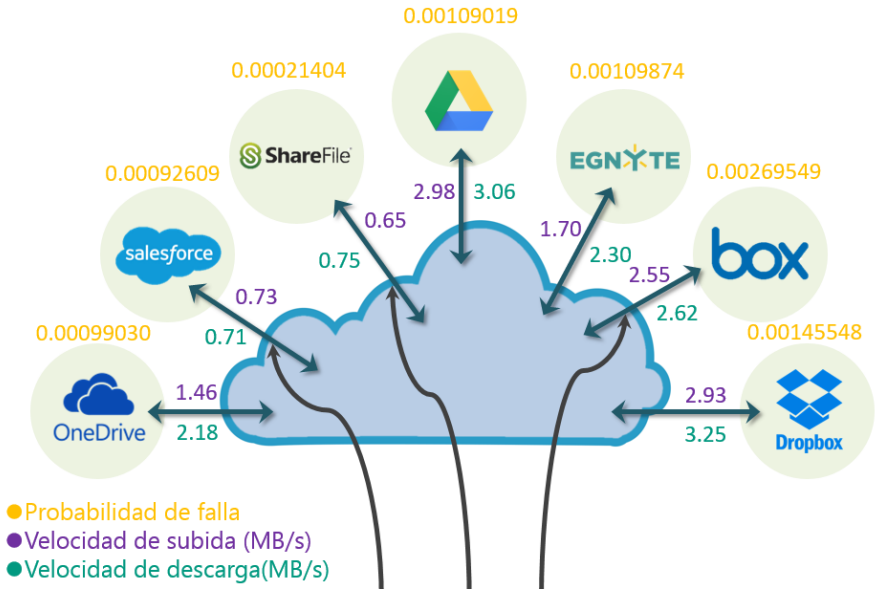

Figura 10. Ambiente multi-nube con parámetros en las nubes.

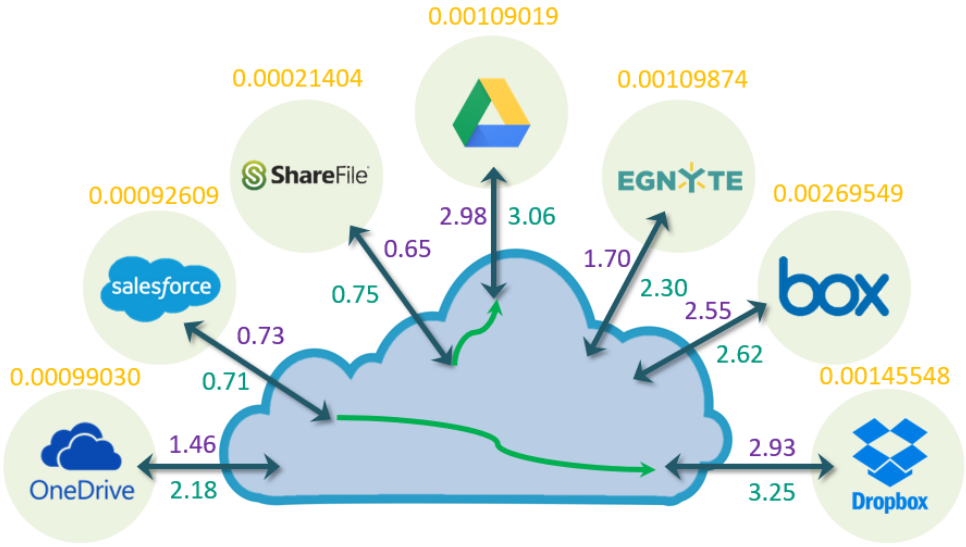

Figura 11. Transferencias de fragmentos entre nubes.

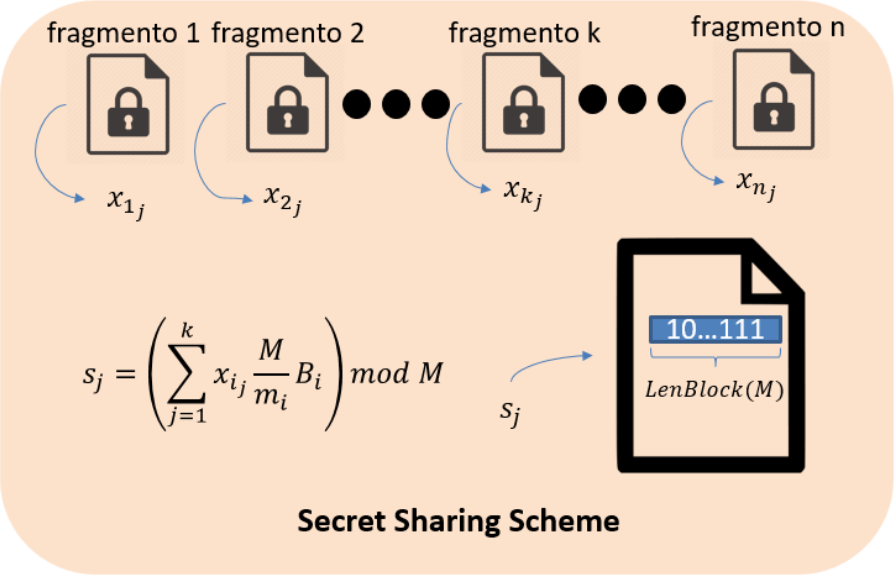

Figura 12. Recuperación de la información a partir de k fragmentos

Por consiguiente, tal y como se muestra en la Figura 12, ya que se descargaron los fragmentos, para recuperar el archivo original, se recorren los fragmentos generados en intervalos de $LenBlock(M)\ mod\ m_i\ \forall\ i \in \{1, \dots, n\}$ para obtener los $x_{i_j}\ \forall j = \{1, \dots, \lceil \frac{D}{LenBlock(M)} \rceil\}$, ya que se obtiene cada uno de los x_{i_j} se utiliza el teorema chino del residuo

$$s_i = \left(\sum_{j=1}^{k} x_{i_j} \frac{M}{m_i} B_i\right) mod\ M \tag{6}$$

Donde B_i es el número tal que $B_i \frac{M}{m_i} + b_i m_i = 1$. Para encontrar B_i se utiliza el algoritmo extendido de Euclides (Cormen et al., 2009) (Algoritmo 2) y dado que el conjunto de módulos m_i son primos relativos por pares, B_i existe y es único.

Algoritmo 2: Extended Euclidean algorithm

1 **procedure** $Extended - Euclid(a, b)$
2 **if** $b == 0$ **then**
3 **return** $(a, 1,0)$
4 **else**(d', x', y') = $Extended - Euclid(b, a\ mod\ b)$
5 $(d', x', y') = (d', y', x' - \lfloor \frac{a}{b} \rfloor y')$
6 **end if**
7 **return** (d, x, y)
8 **end procedure**

3.4.1 Ejemplo de codificación en RNS con números enteros

Sea $\{11,13,17,19\}$ el conjunto de módulos que vamos a utilizar para ejemplificar un esquema (k=2, $n = 4$), sean $m_3 = 17\ y\ m_4 = 19$ los módulos redundantes, esto quiere decir que vamos a utilizar el rango dinámico de los k módulos más pequeños $m_1 = 11\ y\ m_2 = 13$ que es $11 \cdot 13 = 143$.

Sea $X = 3014_{10} = 101111000110_2$ un secreto en su representación en base 10 y base 2, respectivamente. Debido a que solamente podemos representar números dentro del rango dinámico (143), y vamos a trabajar con representaciones en base 2, vamos a establecer el número base 2 previo al rango dinámico que es $2^7 = 128$, entonces vamos a dividir x de derecha a izquierda en segmentos de 7 bits, el segmento $S_1 = 1000110_2 = 70_{10}$ y el segmento $S_2 = 0010111_2 = 23_{10}$. Ahora aplicamos la operación de residuo para la cual utilizamos el símbolo mod, utilizando nuestro conjunto de módulos, sobre $S_1\ y\ S_2$.

$$S_1 mod\ \{11,13,17,19\} = \{4,5,2,13\}_{10} = \{0100,0101,0010,1101\}_2,$$

$$S_2 mod\ \{11,13,17,19\} = \{1,10,6,4\}_{10} = \{0001,1010,0110,0100\}_2.$$

Para crear los fragmentos que van a ser almacenados, vamos a concatenar los residuos base 2 de la siguiente manera

$$C_1 = S_{2,1}S_{1,1} = 00010100_2 = 20_{10}$$

$$C_2 = S_{2,2}S_{1,2} = 10100101_2 = 165_{10}$$

$$C_3 = S_{2,3}S_{1,3} = 01100010_2 = 194_{10}$$

$$C_4 = S_{2,4}S_{1,4} = 01001101_2 = 141_{10}$$

Supongamos, para beneficio de este ejemplo, que los fragmentos 2 y 3 fueron dañados, entonces para recuperar el secreto original, solamente tenemos el fragmento 1 y 4 (C_1, C_4), debido a que utilizamos el esquema $(2,4)$ es posible recuperar el secreto original, si un fragmento más se hubiese dañado, no podríamos recuperar. Para recuperar, vamos a hacer uso del teorema chino del residuo que dice

$$X = \left(\sum_{i=1}^{k} x_i \frac{M}{m_i} B_i \right) mod\ M$$

Para determinar el número B_1 y B_4, vamos a resolver las ecuaciones lineales:

$$\begin{aligned} \tfrac{M}{m_1} B_1 + b_1 m_1 &= 1 \\ \frac{209}{11} B_1 + b_1 11 &= 1 \\ 19(7) + (-12)(11) &= 1 \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{M}{m_4} B_4 + b_4 m_4 &= 1 \\ \frac{209}{19} B_4 + b_4 19 &= 1 \\ 11(7) + (-4)19 &= 1, \end{aligned} \tag{8}$$

Debido a que los módulos que estamos utilizando son primos relativos por pares, los números para resolver las ecuaciones existen y son únicos. Ya que solo contamos con los fragmentos 1 y 4, el rango dinámico ahora es $M = m_1 * m_4 = 209$.

Entonces para reconstruir el secreto original, recordemos que $C_1 = S_{2,1}S_{1,1}$ y $C_4 = S_{2,4}S_{1,4}$, vamos a sumar los residuos pertenecientes a S_1 con los pertenecientes a S_1 y S_2 con S_2, es decir $S_{1,1}$ con $S_{1,4}$ y $S_{2,1}$ con $S_{2,4}$ y tenemos las siguientes ecuaciones

$$S_1 = S_{1,1}b_1M_1 + S_{1,4}b_4M_4$$
$$S_1 = (4 \cdot 7 \cdot 19 + 13 \cdot 7 \cdot 11)\ mod\ 209 \quad (9)$$
$$S_1 = 70_{10} = 1000110_2$$

$$S_2 = S_{2,1}b_1M_1 + S_{2,4}b_4M_4$$
$$S_2 = (1 \cdot 7 \cdot 19 + 4 \cdot 7 \cdot 11)\ mod\ 209 \quad (10)$$
$$S_2 = 23_{10} = 0010111_2$$

Concatenamos S_2 con S_1 para obtener $S_2S_1 = 00101111000110_2 = 3014_{10}$ nuestro secreto original.

3.5 Metodología del análisis

En este trabajo, realizamos un análisis bi-objetivo y multi-objetivo sobre los criterios de velocidad de tiempo de carga (T_{up}), tiempo de descarga (T_{dow}) y la probabilidad de pérdida de información $\Pr(k,n)$. Para esto, formamos tres pares para analizar, T_{up} con $\Pr(k,n)$, T_{dow} con $\Pr(k,n)$ y T_{up} con T_{dow}. Debido a que el par de T_{up} con T_{dow} no se encuentra en conflicto, diseñamos una estrategia para minimizar ambos criterios pero no se continuó con el análisis a profundidad.

Para facilitar la visualización del análisis bi-objetivo, se realiza una normalización por medio del método min-max para colocar todos los valores dentro del intervalo [0,1], dónde 0 representa el mejor valor encontrado y 1 el peor.

$$\gamma = \frac{strategy\ metric\ value - worst\ metric\ value}{(best\ metric\ value - worst\ metric\ value)} \quad (11)$$

Para realizar el análisis multi-objetivo, hacemos uso de una métrica conocida como Distancia Generacional, el valor obtenido por esta métrica representa qué tan alejado se encuentra un frente aproximado de Pareto conocido ($PF_{conocido}$) a uno verdadero ($PF_{verdadero}$) o de referencia ($PF_{referencia}$). Se define como:

$$G \triangleq \left(\frac{\sum_{i=1}^{n} d_i^p}{n}\right)^{\frac{1}{p}}, \tag{12}$$

dónde n es el número de elementos en $PF_{conocido}$, p=2 y d_i es la distancia Euclidiana del elemento i en $PF_{conocido}$ al elemento más cercano de $PF_{referencia}$. Si $G = 0$, $PF_{conocido} = PF_{referencia}$, cualquier otro caso indica que $PF_{conocido}$ se desvía de $PF_{referencia}$ (Van Veldhuizen & Lamont, 2000).

Capítulo 4. Resultados

4.1 Configuración experimental

Para poder realizar un análisis experimental de las estrategias propuestas, es necesario caracterizar cada una de las nubes utilizadas por el sistema. Para esto, elegimos siete proveedores de almacenamiento en la nube reales y realizamos experimentos para obtener su velocidad de carga y de descarga, también su probabilidad de negación de acceso.

En esta sección se presenta el análisis experimental, las velocidades de acceso y probabilidad de negación de acceso de cada proveedor de servicio en la nube. El sistema en el cuál se efectuaron los experimentos fue desarrollado utilizando el lenguaje de programación Java. Los experimentos se ejecutaron en un servidor Express x3650 M4, con dos procesadores Xeon IbyBridge E5-2650v2 95W @ 2.6 GHz y una velocidad de internet simétrica de 1 Gbps. Con CentOS Linux 7.1.1503 como sistema operativo.

4.1.1 Medición de velocidades con Kloudless

Debido a que el sistema propuesto utiliza múltiples nubes de almacenamiento, el primer paso que dimos fue investigar qué herramientas se encontraban a nuestra disposición que nos permitieran centralizar el manejo de cada una de las nubes.

Después de analizar las ventajas y desventajas de productos como jclouds (https://jclouds.apache.org), apigee (https://apigee.com), rightscale (https://www.rightscale.com) y kloudless (https://kloudless.com), decidimos que la herramienta cuya oferta se aproximaba más a lo que necesitábamos era Kloudless (Figura 8).

Kloudless fue fundada en agosto de 2011 por cuatro alumnos de la Universidad de California en Berkeley, Eliot Sun, Timothy Liu, Vinod Chandru y Brian Tang. Finalista del Techcrunch Disrupt 2013 en Nueva York, ofrece un API universal para almacenamiento en la nube. De acuerdo con su página de internet, cuentan con más de veinte conectores a proveedores de almacenamiento en la nube y se encuentran en constante trabajo para agregar más.

Figura 13. Proveedores de almacenamiento disponibles en Kloudless. (https://kloudless.com/file-storage-api)

La característica que ofrece Kloudless que más nos interesó, y el factor principal por el cuál utilizamos su API, es la de transferir archivos de un proveedor a otro. Para que el sistema que proponemos tenga sentido, no es posible que si se desea cambiar un fragmento de generado por el sistema de compartición de secretos de una nube a otra, el usuario tenga que descargar el fragmento completo a su ordenador y después cargarlo a la nube del proveedor deseado.

Primero, seleccionamos los proveedores que íbamos a utilizar. El primer criterio de selección fue que tuvieran una versión gratuita y el segundo es que tuvieran una versión de prueba también gratuita sin la necesidad de ingresar datos de algún método de pago. Los proveedores que cumplieron con esos criterios fueron Google Drive, Dropbox, Box, Sharefile, OneDrive, Egnyte y Salesforce. Ya que se implementaron los puntos de acceso para cargar y descargar archivos con esos proveedores, durante tres días utilizamos el API de Kloudless para cargar y descargar un archivo de 50 MB a cada una de las nubes cada hora.

Contrario a nuestras expectativas, los tiempos y velocidades de carga (Tabla 4) y descarga (Tabla 5) de las nubes presentaron similitud. La diferencia entre la nube con mayor velocidad de carga promedio y la de menor velocidad de carga promedio es de 0.33 MBps, mientras que en descarga la diferencia es aún menor con 0.2 MBps. Esto podría indicar que existe un cuello de botella en el ancho de banda, el flujo de datos permitidos por el API o el tamaño del archivo de prueba es muy pequeño. Para descartar la primera

opción, realizamos pruebas de velocidad con la herramienta iPref en el servidor que ejecuta las pruebas y obtuvimos velocidades cercanas a 1 Gbps. Para descartar la tercera opción, se realizaron pruebas cortas con archivos de 100 y 150 MB y los resultados obtenidos eran similares al utilizar el archivo de 50 MB. Por lo tanto, llegamos a la conclusión de que el flujo de datos permitido por Kloudless en carga y descarga era limitado.

Tabla 4. Tiempo y velocidad de carga.

Proveedor	**Carga**						
	Min (s)	**Max (s)**	**Promedio (s)**	**σ**	**Min (MBps)**	**Max (MBps)**	**Promedio (MBps)**
Box	38.79	52.47	40.80	1.76	0.95	1.29	1.23
Google Drive	38.80	51.65	40.83	2.12	0.97	1.29	1.22
Salesforce	38.64	54.62	41.12	2.66	0.92	1.29	1.22
Egnyte	38.65	86.59	42.47	6.46	0.58	1.29	1.18
DropBox	41.81	55.34	44.40	2.30	0.90	1.20	1.13
One Drive	41.82	88.63	45.57	6.37	0.56	1.20	1.10
Sharefile	41.40	98.50	55.58	15.72	0.51	1.21	0.90

Tabla 5. Tiempo y velocidad de descarga.

Proveedor	**Descarga**						
	Min (s)	**Max (s)**	**Promedio (s)**	**σ**	**Min (MBps)**	**Max (MBps)**	**Promedio (MBps)**
Box	38.73	58.58	41.34	3.79	0.85	1.29	1.21
Google Drive	38.48	95.34	42.33	8.42	0.52	1.30	1.18
One Drive	39.00	63.75	42.48	4.16	0.78	1.28	1.18
DropBox	38.79	86.16	42.77	7.00	0.58	1.29	1.17
Salesforce	39.05	92.17	42.96	8.49	0.54	1.28	1.16
Sharefile	39.76	86.89	43.53	6.08	0.58	1.26	1.15
Egnyte	38.43	85.42	48.66	11.93	0.59	1.30	1.03

Durante tres días utilizamos el API de Kloudless para mover un archivo de 50 MB de una nube al resto de las nubes cada hora. A diferencia de los resultados obtenidos en carga y descarga, la transferencia de datos entre nubes sí presentó la variabilidad que nosotros suponíamos, tal y como se muestra de la Tabla 6 a la 12. En promedio, las transferencias más rápidas son aquellas cuya salida son de la nube de One Drive y las menos veloces, son las que tienen salida desde Sharefile. El proveedor con la menor variabilidad en la velocidad de transferencia brindada es Salesforce y el de mayor variabilidad es Sharefile, con esto se puede concluir que en transferencia de datos de una nube a otra, Sharefile presenta el peor rendimiento ya que cuenta con la menor velocidad y la mayor variabilidad.

Tabla 6. Velocidad de transferencia desde Box.

Proveedor	Box (MBps)		
	Min	Max	Promedio
One Drive	1.58	5.43	3.61
Google Drive	2.51	7.63	5.97
ShareFile	0.63	5.79	2.52
Egnyte	5.31	7.97	6.51
Salesforce	3.46	4.28	3.96
Dropbox	1.20	5.7	4.40

Tabla 7. Velocidad de transferencia desde One Drive.

Proveedor	One Drive (MBps)		
	Min	Max	Promedio
Box	2.24	7.65	5.18
Google Drive	3.91	9.62	6.67
ShareFile	0.71	5.70	2.49
Egnyte	4.02	8.46	6.62
Salesforce	2.72	4.211	3.77
DropBox	1.83	4.84	3.89

Tabla 8. Velocidad de transferencia desde Google Drive.

Proveedor	Google Drive (MBps)		
	Min	Max	Promedio
Box	1.65	6.93	3.65
One Drive	1.36	7.45	4.79
ShareFile	1.06	5.95	3.02
Egnyte	5.05	9.04	7.45
Salesforce	1.36	4.02	3.36
DropBox	7.82	5.10	6.93

Tabla 9. Velocidad de transferencia desde ShareFile.

Proveedor	ShareFile (MBps)		
	Min	Max	Promedio
Box	0.60	3.59	2.61
One Drive	3.04	3.04	2.36
Google Drive	1.33	3.70	2.72
Egnyte	3.50	1.55	2.75
Salesforce	0.96	3.81	2.79
DropBox	1.63	2.84	2.39

Tabla 10. Velocidad de transferencia desde Egnyte.

Proveedor	Egnyte (MBps)		
	Min	Max	Promedio
Box	2.57	3.0	3.22
One Drive	2.37	2.82	2.62
Google Drive	1.76	3.47	2.99
ShareFile	0.55	2.96	2.04
Salesforce	2.91	3.46	3.20
DropBox	0.79	2.38	2.06

Tabla 11. Velocidad de transferencia desde Salesforce.

Proveedor	Salesforce (MBpss)		
	Min	Max	Promedio
Box	1.41	3.39	3.10
One Drive	1.32	2.83	2.43
Google Drive	2.86	3.25	3.06
ShareFile	1.18	2.92	2.37
Egnyte	3.02	3.34	3.19
DropBox	2.51	2.95	2.80

Tabla 12. Velocidad de transferencia desde Dropbox.

Proveedor	Dropbox (MBpss)		
	Min	Max	Promedio
Box	0.91	7.29	4.28
One Drive	0.85	5.18	3.36
Google Drive	5.28	8.57	6.77
ShareFile	1.34	5.58	3.01
Egnyte	5.22	7.97	6.62
Salesforce	3.16	4.03	3.80

4.1.2 Medición de velocidades con APIs individuales

Después analizar los resultados obtenidos al utilizar el API de Kloudless, llegamos a la conclusión de que no era posible que las velocidades de carga y descarga de cada una de las nubes fueran tan similares, más aún la transferencia de archivos entre las nubes sí presentaba diferencia en velocidades, además el ancho de banda utilizado es bastante superior al reflejado en los resultados obtenidos.

Por los motivos mencionados en el párrafo anterior, se creó un sistema de transferencia utilizando los APIs que ofrecen los siguientes siete proveedores: Google Drive, Dropbox, Box, Sharefile, OneDrive, Egnyte y Salesforce.

Para crear la conexión con Google Drive, Dropbox, Box y Sharefile, se hizo uso de la librería en Java que ellos ofrecen. Para OneDrive, Egnyte y Salesforce, debido a que sólo ofrecen puntos de acceso mediante un API REST, se utilizó la librería en Java de Apache HttpClient (https://hc.apache.org/) para hacer la conexión.

Durante tres días, cada hora, se cargó y se descargó un archivo de 200 MB a cada proveedor de almacenamiento en la nube. A diferencia de los resultados obtenidos al utilizar el API de Kloudless, realizar la medición de velocidades con el API proporcionado por cada proveedor (Tabla 13), presentó la variabilidad que suponíamos. La diferencia entre la nube con mayor velocidad de carga promedio y la de menor velocidad de carga promedio es de 2.47 MBps, mientras que en descarga la diferencia es aún mayor con 2.54 MBps, números muy diferentes a los obtenidos por utilizar el API de Kloudless para cargar y descargar archivos.

Tabla 13. Velocidades de carga u_j y descarga d_j de los siete proveedores.

Proveedor	Velocidad de carga u_j (MBps)			Velocidad de descarga d_j (MBps)		
	Min	Max	Promedio	Min	Max	Promedio
GoogleDrive	1.79	3.24	2.98	2.15	3.26	3.06
OneDrive	0.91	1.70	1.46	1.21	2.41	2.18
Dropbox	2.59	3.05	2.93	3.07	3.32	3.25
Box	1.91	3.26	2.55	2.01	3.20	2.62
Egnyte	1.24	1.93	1.70	2.17	2.36	2.30
Sharefile	0.11	0.65	0.51	0.72	0.76	0.75
Salesforce	0.52	0.73	0.64	0.68	0.72	0.71

4.1.3 Probabilidad de falla de una nube

Suponemos como falla de un proveedor en la nube el no poder hacer uso de un archivo almacenado. También, que la probabilidad de falla es directamente proporcional al tamaño del archivo. Mientras más grande sea el archivo, más alta es la probabilidad de que dicho archivo se encuentre una sección dañada del disco duro.

Debido a que no es posible obtener directamente de un proveedor cuánto tiempo falló su servicio de almacenamiento, utilizamos el análisis presentado por CloudHarmony en 2015 (Butler, 2015). CloudHarmony monitorea 91 proveedores de servicio en la nube de los cuales 21 proveen almacenamiento. Algunos de esos 21 se describen a continuación.

Alibaba Cloud (https://alibabacloud.com/). Es una sección de la compañía Alibaba Group. Alibaba Cloud ofrece una gran variedad de servicios de cómputo, entre ellos el almacenamiento. En cuestión de seguridad, ofrecen protección Anti-DDos, transferencia de datos por SSL y encriptación estacionaria con AES 256-bits. También, dicen tener un 99.9% de disponibilidad, para respaldar este dato, manejan una replicación triple de cada objeto almacenado. Una de sus desventajas principales es que el ancho de banda está restringido a 5 Gbps fuera de China.

CenruryLink Cloud (https://ctl.io). CenturyLink es una compañía de telecomunicaciones fundada en el año 1930. En 2011 adquiere a Savvis, Inc. que un año más tarde pasaría a ser la división de servicios

en la nube de CenturyLink colocando los cimientos de CenturyLink Cloud. A pesar de que no ofrecen encriptación estacionaria, cuentan con una gran variedad de servicios de terceros soportados que podrían ser suficientes. Para atacar el problema de disponibilidad ofrecen replicación y redundancia.

Rackspace Cloud (https://rackspace.com/cloud). Rackspace inició como una compañía dedicada al alojamiento de páginas web en 1998. Entre los años 2006 y 2008, comenzó a ofrecer servicios de cómputo en la nube. Todo su tráfico de información se transmite por canales SSL. Sin embargo, no cuentan con sistemas de encriptación, detección de virus o compresión de objetos que entran o salen. Los archivos que son almacenados en su nube, se replican tres veces y se distribuyen en distintas secciones del mismo centro de datos.

La explicación de cómo se realiza el monitoreo, lo indican en su página web bajo la sección de monitoreo de servicio: “La disponibilidad o el tiempo activo de un servicio es una característica crítica del rendimiento de la nube. A menudo se supone que los proveedores de la nube tienen medidas para proporcionar una disponibilidad muy alta. Para medir el tiempo de actividad, tenemos servicios configurados con la mayoría de los proveedores de nube pública y tenemos monitores instalados para rastrear cualquier interrupción de esos servicios. Normalmente realizamos un seguimiento con el proveedor después de una interrupción prolongada y anexamos comentarios de respaldo al informe de tiempo de actividad. También excluimos el mantenimiento programado. Hemos encontrado muy poca correlación entre los SLA y el tiempo de actividad real. Desde que comenzamos a monitorear los servicios alrededor de enero de 2010, hemos observado variaciones significativas entre la disponibilidad de diferentes servicios en la nube” (https://cloudharmony.com/services).

Lamentablemente para nuestro trabajo, no todos los proveedores de servicio que utilizamos en nuestro análisis de velocidades, fueron monitoreados por CloudHarmony en el 2015. Por lo tanto, nos vimos en la necesidad de hacer las siguientes suposiciones. Debido a que el análisis muestra que el mejor proveedor de servicio solo tuvo 34 minutos de interrupción de acceso, presentando una disponibilidad de 99.99 durante el año 2015 y el peor de los proveedores monitoreados tuvo 31 horas y 29 minutos de interrupción de acceso, presentando una disponibilidad de 99.64. Suponemos que los valores reales para los proveedores de servicio en la nube que se manejan en este trabajo, no se encuentran fuera de ese intervalo. Por lo tanto, tomamos algunos de los valores reportados por CloudHarmony y los asignamos de forma arbitraria como valores máximos de probabilidad de falla a las nubes que se manejan en este trabajo, la probabilidad mínima de falla, la tomamos como la mitad del valor asignado.

Dado que la probabilidad de falla es directamente proporcional al tamaño del archivo. Calculamos las probabilidades de cada proveedor de servicio utilizando su valor mínimo y máximo de probabilidad de falla y un rango de tamaño de archivo de 10 MB a 200 MB.

Tabla 14. Probabilidad de falla de cada nube

Proveedor	Probabilidad de falla		
	Min	Max	Promedio
Sharefile	0.000142695	0.0002854	0.00021404
Salesforce	0.000617390	0.0012348	0.00092609
One Drive	0.000660200	0.0013204	0.00099030
Google Drive	0.000726790	0.0014536	0.00109019
Egnyte	0.000732495	0.0014650	0.00109874
DropBox	0.000970320	0.0019406	0.00145548
Box	0.001796995	0.0035940	0.00269549

En la Tabla 14 se muestran los valores de probabilidad de falla que utilizamos para los proveedores y en la Figura 14 se ilustra su comportamiento lineal. Con esto, le asignamos una ecuación de la recta a cada proveedor para poder obtener la probabilidad de falla utilizando cualquier tamaño de archivo de entrada. Es importante resaltar que la probabilidad de falla máxima no cambia, solo la pendiente de la recta.

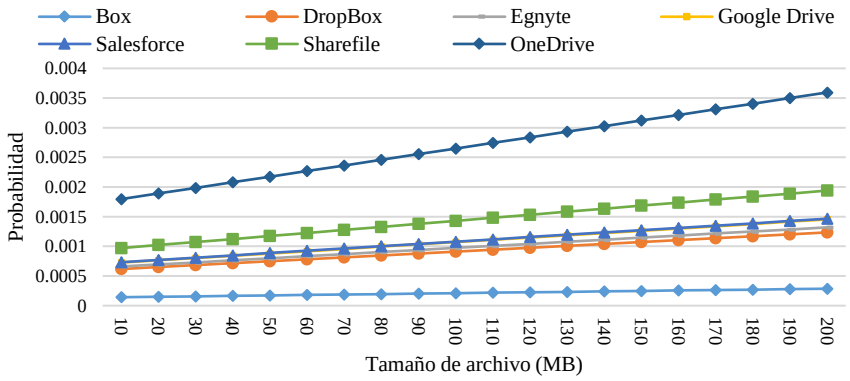

Figura 14. Probabilidad de falla de los siete proveedores.

4.2 Resultados experimentales

Para el primer experimento, utilizamos las estrategias BestUpload, BestDownload y AdaptiveSpeed con las velocidades obtenidas al utilizar el API de Kloudless (Sección 4.1.1), el modelo matemático definido en la sección 3.2 y el sistema descrito en la Sección 3.4, realizamos una comparativa de los resultados.

Figura 15. Tiempo de descarga por (k, n) en segundos

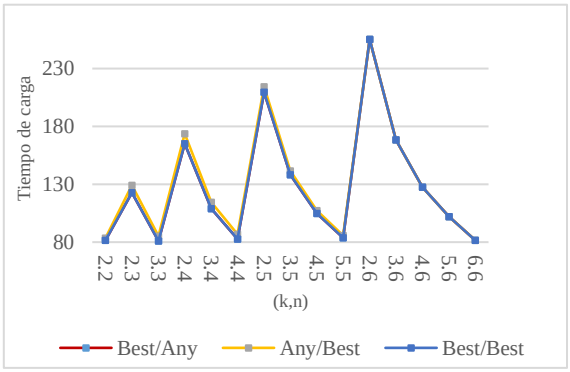

Figura 16. Tiempo de carga por (k, n) en segundos

En la Figura 15 se alcanza a apreciar la diferencia de tiempo de descarga entre utilizar la estrategia AdaptiveSpeed y utilizar alguna de las otras dos estrategias. La Figura 16 podría no ser muy ilustrativa ya que es complicado interpretar si existe deferencia alguna en ciertos casos. Para esto las figuras 17 y 18, ilustran la diferencia en velocidades. La velocidad más alta se presenta cuando se utilizan las configuraciones (2,2), (3,3), (4,4), (5,5) y (6,6), esto debido a que las configuraciones de la forma (n, n) presentan la menor redundancia. Sin embargo, como se muestra en la Figura 19, dichas configuraciones cuentan con la mayor probabilidad de falla.

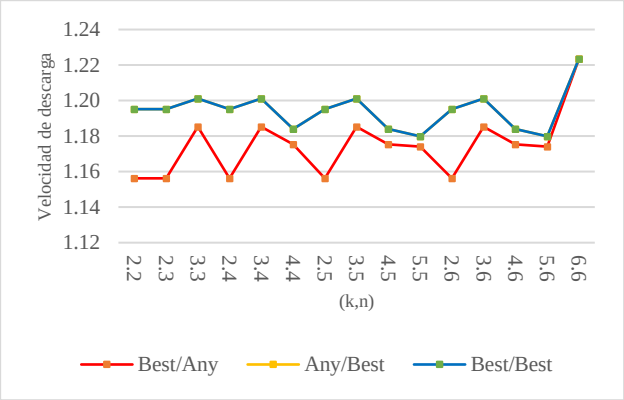

Figura 17. Velocidad de descarga MBps por (k, n)

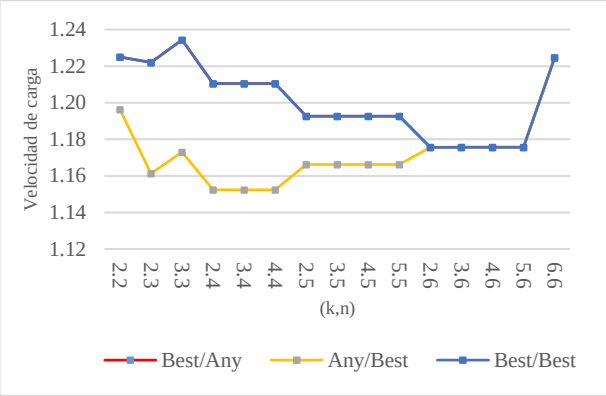

Figura 18. Velocidad de carga MBps por (k, n)

En la Tabla 15 se presenta la reducción en tiempo de carga al utilizar la estrategia AdaptiveSpeed (AS) contra la estrategia BestUpload (BU) y la reducción en tiempo de descarga al utilizar la estrategia AdaptiveSpeed contra utilizar la estrategia BestDownload (BD). De la misma manera, en la Tabla 16 se presenta el incremento en velocidad. En ambas tablas, la mejora en porcentaje no supera el 6%. Esto se debe a que las diferencias de velocidades entre las nubes es muy poca.

Tabla 15. Reducción de tiempo de carga y descarga entre estrategias.

(k,n)	AS vs BU reducción en tiempo de descarga %	AS vs BD reducción en tiempo de carga %
(2,2)	3.25	2.34
(2,3)	3.25	4.96
(3,3)	1.31	4.96
(2,4)	3.25	4.81
(3,4)	1.31	4.81
(4,4)	0.73	4.81
(2,5)	3.25	2.21
(3,5)	1.31	2.21
(4,5)	0.73	2.21
(5,5)	0.49	2.21
(2,6)	3.25	0.00
(3,6)	1.31	0.00
(4,6)	0.73	0.00
(5,6)	0.49	0.00
(6,6)	0.00	0.00

Tabla 16. Incremento de velocidad de descarga y carga entre estrategias.

(k,n)	AS vs BU incremento de velocidad de descarga %	AS vs BD incremento de velocidad de carga %
(2,2)	3.36	2.39
(2,3)	3.36	5.22
(3,3)	1.33	5.22
(2,4)	3.36	5.05
(3,4)	1.33	5.05
(4,4)	0.73	5.05
(2,5)	3.36	2.26
(3,5)	1.33	2.26
(4,5)	0.73	2.26
(5,5)	0.50	2.26
(2,6)	3.36	0.00
(3,6)	1.33	0.00
(4,6)	0.73	0.00
(5,6)	0.50	0.00
(6,6)	0.00	0.00

Realizamos un segundo experimento en el que utilizamos las velocidades de las nubes que obtuvimos al implementar los APIs proporcionados por cada proveedor (Sección 4.1.2) y además la probabilidad de falla (Sección 4.1.3), el modelo matemático definido en la sección 3.2 y el sistema descrito en la Sección 3.4.

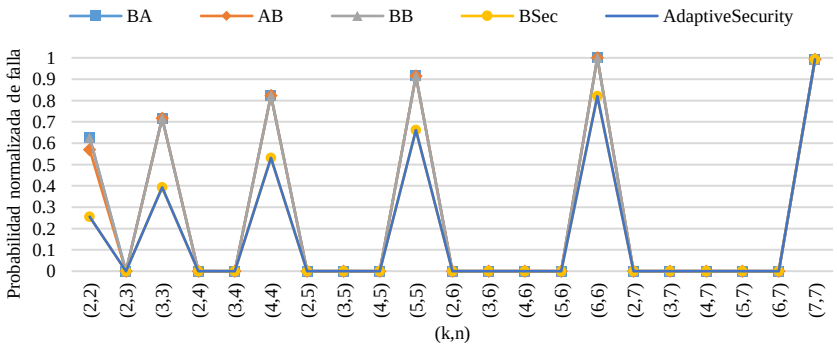

Figura 19. Probabilidad normalizada de falla del sistema por (k, n).

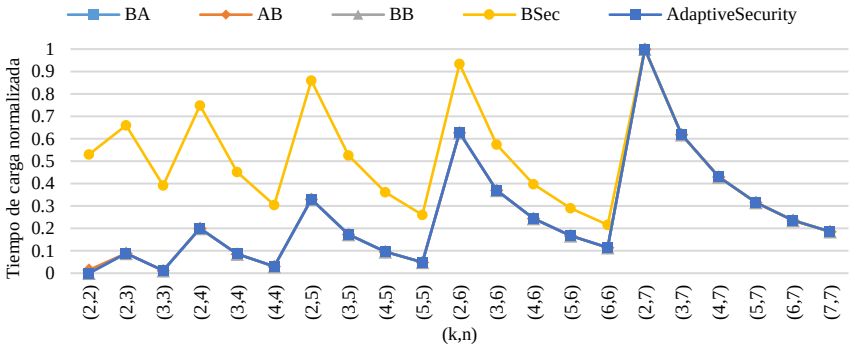

Figura 20. Tiempo de carga por (k, n) normalizado.

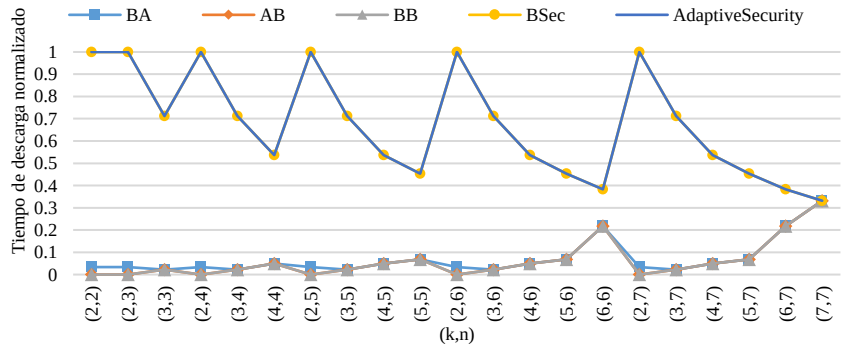

Figura 21. Tiempo de descarga por (k, n) normalizado.

En las figuras 19, 20 y 21 se aprecia el conflicto que se genera en los criterios de probabilidad de falla del sistema, tiempo de descarga y tiempo de carga al utilizar alguna configuración (k, n). Para probabilidad de falla, las configuraciones que generan las peores soluciones son las de la forma (n, n) mientras que para ambos tiempos, dichas configuraciones generan las mejores soluciones. Además, la estrategia que brinda el mejor resultado para el criterio de probabilidad de falla es la menos adecuada para el criterio de tiempo de descarga.

En la Tabla 17, primera columna, se muestra el porcentaje de mejora en probabilidad de perdida al utilizar AdaptiveSecurity en lugar de BestUpload, la configuración (k, n) que presenta la mayor mejora es $(2,2)$ con 22.86% y la de menor mejora $(2,6)$ 9.77E-13%, excluyendo las configuraciones donde $n = 7$

ya que se utilizan las mismas nubes para ambas estrategias. Debido a que $(2,6)$ es una configuración que tiene un muy buen desempeño en probabilidad de pérdida de información, la mejora es mínima, caso contrario para $(2,2)$.

En la segunda columna, se encuentra la mejora en tiempo de carga al utilizar AdaptiveSecurity en lugar de BestSecurity, la configuración (k,n) que presenta la mayor mejora es $(2,2)$ con 52.99% y la de menor mejora $(6,6)$ 9.12%, excluyendo las configuraciones donde $n = 7$ ya que se utilizan las mismas nubes para ambas estrategias.

En la tercera columna, se encuentra la mejora en tiempo de descarga al utilizar AdaptiveSecurity en lugar de BestDownload, los valores son todos negativos ya que en lugar de obtener una mejora hay una disminución en rendimiento que en seis configuraciones llega a ser hasta el doble.

Tabla 17. Reducción de probabilidad de pérdida de información, tiempo de carga y descarga con AdaptiveSecurity. (%)

(k,n)	**AdaptiveSecurity vs**		
	BestUpload probabilidad de pérdida	**BestSecurity tiempo de carga**	**BestDownload tiempo de descarga**
(2,2)	22.86281	52.99145	-100
(2,3)	0.02201	52.53054	-100
(3,3)	18.90773	37.52345	-67.54177
(2,4)	9.56E-06	45.64873	-100
(3,4)	0.022836	33.56643	-67.54177
(4,4)	16.00884	26.63594	-46.51163
(2,5)	3.35E-09	39.82869	-100
(3,5)	9.96E-06	30.04049	-67.54177
(4,5)	0.022816	24.17678	-46.51163
(5,5)	13.2764	20.19928	-36.07306
(2,6)	9.77E-13	18.91419	-100
(3,6)	3.1E-09	14.89951	-67.54177
(4,6)	9.35E-06	12.36641	-46.51163
(5,6)	0.018882	10.57771	-36.07306
(6,6)	8.976775	9.12191	-13.62726
(2,7)	0	0	-100
(3,7)	0	0	-67.54177
(4,7)	0	0	-46.51163
(5,7)	0	0	-36.07306
(6,7)	0	0	-13.62726
(7,7)	0	0	0

4.2.1 Cambios de parámetros en el ambiente

Para simular el impacto que pueda llegar a tener la variabilidad de parámetros en un ambiente multi-nube, analizamos 21 configuraciones distintas y seis estrategias durante 60 simulaciones con distintos valores de velocidad de carga, velocidad de descarga y probabilidad de falla del sistema en cada simulación. Los valores para las nubes se seleccionaron con una distribución uniforme entre el valor máximo y el mínimo obtenido para cada nube en las Secciones 4.1.2 y 4.1.3.

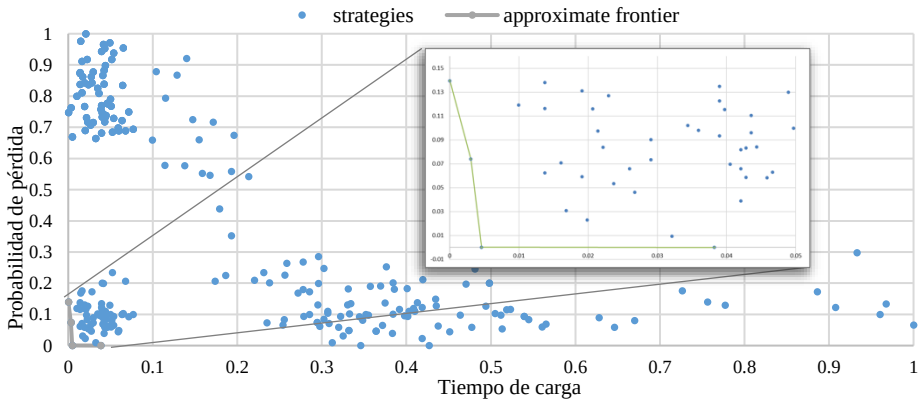

Figura 22. Aproximación de Pareto para la configuración $(3,5)$. Probabilidad de pérdida de información vs tiempo de carga

En la Figura 22, se tiene el espacio de solución para la configuración (3,5) para probabilidad de pérdida de información y tiempo de carga. Cada punto representa una estrategia distinta en una de las 60 simulaciones y el frente aproximado de Pareto está compuesto por 4 soluciones de la estrategia AdaptiveSecurity (Tabla 18) lo cual indica que a pesar de que los parámetros varíen, la estrategia AdaptiveSecurity da los mejores resultados en tiempo de carga y probabilidad de pérdida de información para la configuración (3,5).

Tabla 18. Miembros del frente aproximado de Pareto de la configuración $(3,5)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Probabilidad de pérdida de información
AdaptiveSecurity	0.004587	0.139608
AdaptiveSecurity	0.003058	0.073982
AdaptiveSecurity	0.004587	0.000166
AdaptiveSecurity	0.038226	0

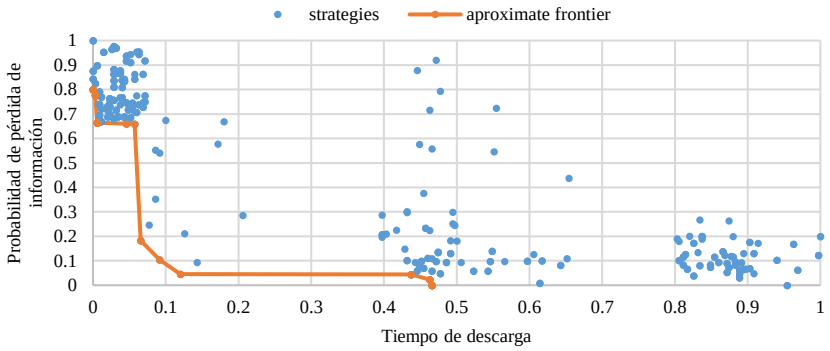

Figura 23. Aproximación de Pareto para la configuración $(3,5)$. Probabilidad de pérdida de información vs tiempo de descarga

En la Figura 23, se encuentra el espacio de solución para la configuración (3,5) para probabilidad de pérdida de información y tiempo de descarga. Cada punto representa una estrategia distinta en una de las 60 simulaciones y el frente aproximado de Pareto está compuesto por 5 soluciones de la estrategia Random, 3 de BestSecurity, 3 de AdaptiveSecurity, 3 de BestUpload y 3 de AdaptiveSpeed (Tabla 19). Lo cual indica que para la configuración $(3,5)$ en probabilidad de pérdida de información y tiempo de descarga no contamos con una estrategia que domine al resto.

Tabla 19. Miembros del frente aproximado de Pareto de la configuración $(3,5)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de descarga normalizado	Probabilidad de pérdida de información normalizada
BestSecurity	0.465714	0
AdaptiveSecurity	0.465714	0
BestSecurity	0.462857	0.022908
AdaptiveSecurity	0.462857	0.022908
BestSecurity	0.437143	0.044249
AdaptiveSecurity	0.437143	0.044249
Random	0.120000	0.046028
Random	0.091429	0.103979
Random	0.065714	0.182261
Random	0.057143	0.659180
Random	0.045714	0.660219
BestUpload	0.005714	0.664375
AdaptiveSpeed	0.005714	0.664375
BestUpload	0.002857	0.777316
AdaptiveSpeed	0.002857	0.777316
BestDownload	0	0.799591

En las tablas 24-44, se encuentran los miembros del frente aproximado de Pareto de cada una de las estrategias para los objetivos de tiempo de carga y probabilidad de pérdida de información, además en las figuras 26-46 se presentan los frentes aproximados. De estos frentes aproximados, la estrategia AdaptiveSecurity aparece como miembro 51.70% de las ocasiones, además de ser la estrategia con más miembros en el frente aproximado en 15 de las 21 configuraciones (Tabla 20). Por lo tanto, podemos concluir que la estrategia AdaptiveSecurity tiene el mejor rendimiento para los criterios de tiempo de carga y probabilidad de pérdida de información independientemente de la configuración.

Tabla 20. Cantidad de miembros en el frente aproximado para tiempo de carga y probabilidad de pérdida de información por configuración.

(k,n)	AdaptiveSecurity	Random	AdaptiveSpeed	BestUpload	BestDownload	BestSecurity
(2,2)	2	1	0	0	0	0
(2,3)	2	1	0	0	0	0
(3,3)	3	0	0	0	0	0
(2,4)	3	2	0	0	0	0
(3,4)	4	0	0	0	0	0
(4,4)	5	0	0	0	0	0
(2,5)	4	0	0	0	0	0
(3,5)	4	0	0	0	0	0
(4,5)	4	0	0	0	0	0
(5,5)	4	0	0	0	0	0
(2,6)	5	0	0	0	0	0
(3,6)	7	0	0	0	0	0
(4,6)	5	0	0	0	0	0
(5,6)	5	0	0	0	0	0
(6,6)	5	0	0	0	0	0
(2,7)	3	2	1	1	1	3
(3,7)	1	4	2	2	2	1
(4,7)	3	2	3	3	2	3
(5,7)	4	2	3	3	2	4
(6,7)	4	5	1	1	1	4
(7,7)	2	1	2	2	4	2

En las tablas 45-65, se encuentran los miembros del frente aproximado de Pareto de cada una de las estrategias para los objetivos de tiempo de descarga y probabilidad de pérdida de información además en las figuras 47-67 se presentan los frentes aproximados. De estos frentes aproximados, la estrategia AdaptiveSpeed aparecen como miembro 89 de las 404 ocasiones, al igual que BestDownload, siendo estás dos estrategias las más dominantes en la cantidad de miembros de los frentes aproximados. Por lo tanto, podemos concluir que las estrategias AdaptiveSpeed y BestDownload tienen el mejor rendimiento para los criterios de tiempo de descarga y probabilidad de pérdida de información independientemente de la configuración.

Tabla 21. Cantidad de miembros en el frente aproximado para tiempo de descarga y probabilidad de pérdida de información por configuración.

(k,n)	AdaptiveSecurity	Random	BestUpload	BestDownload	AdaptiveSpeed	BestSecurity
(2,2)	2	5	3	4	4	2
(2,3)	5	7	0	3	3	5
(3,3)	11	3	4	6	6	11
(2,4)	1	7	0	3	3	1
(3,4)	4	6	4	2	2	4
(4,4)	7	4	5	5	5	7
(2,5)	2	4	0	3	3	2
(3,5)	3	5	2	1	1	3
(4,5)	4	3	2	2	2	4
(5,5)	5	1	8	7	7	5
(2,6)	3	3	0	3	3	3
(3,6)	1	3	1	5	5	1
(4,6)	1	3	1	3	3	1
(5,6)	2	2	1	9	9	2
(6,6)	4	0	1	8	8	4
(2,7)	1	1	0	5	5	1
(3,7)	0	1	3	2	2	0
(4,7)	0	0	2	3	3	0
(5,7)	1	0	5	5	5	1
(6,7)	0	1	0	7	7	0
(7,7)	3	4	1	3	3	3

Debido a que el análisis en dos objetivos puede no ser suficiente, realizamos un análisis de tres objetivos (en tres dimensiones) sobre las estrategias. Para esto, utilizamos la métrica de distancia generacional presentada en la sección 3.4. Primero para obtener $PF_{referencia}$, creamos un espacio con todas las soluciones que fueron encontradas durante el experimento y encontramos la malla para ese conjunto de soluciones, en realidad no podemos saber si esa malla es la Pareto óptima, pero que consideramos que es lo más a $PF_{verdadero}$cercano que podemos encontrar (Figura 24). Después un $PF_{conocido}$ está compuesto por la malla aproximada que encontramos al utilizar un espacio de soluciones solamente de una configuración y agrupar las estrategias dentro de ese espacio.

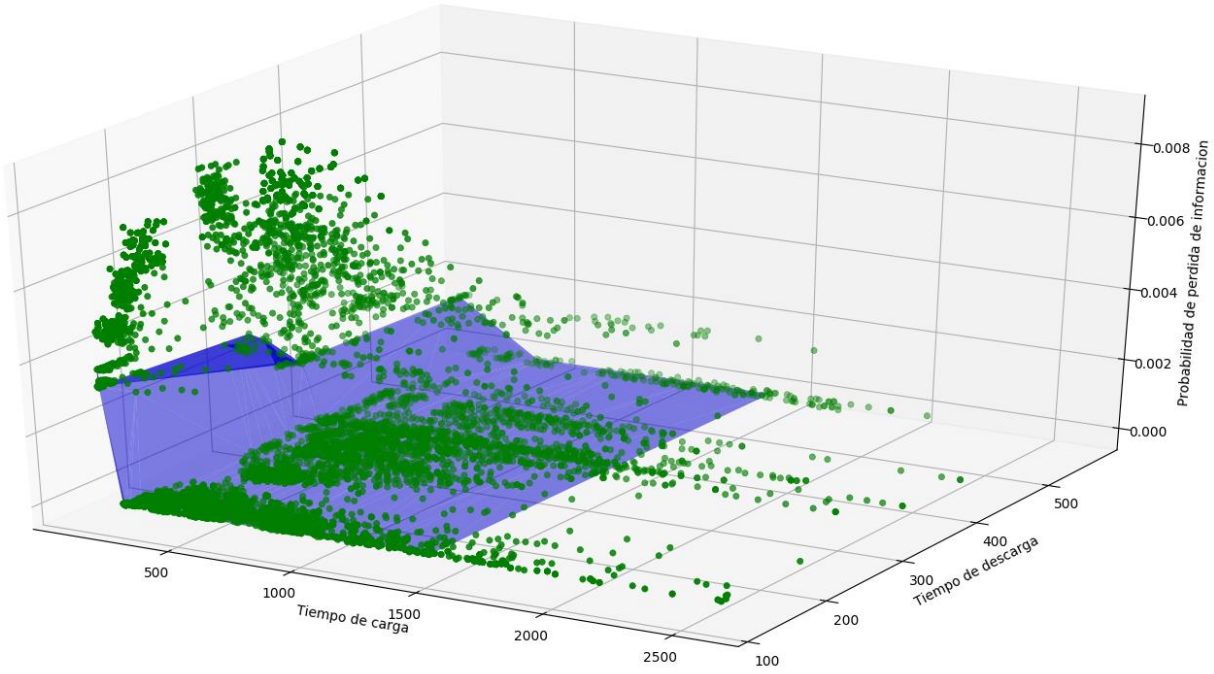

Figura 24. Espacio de todas las soluciones encontradas y malla artificial

En la Figura 25, se muestra el espacio de soluciones de la configuración (3,5) y la malla aproximada de la estrategia AdaptiveSecurity (azul) y AdaptiveSpeed (rojo). Por cada estrategia, se encuentra una malla similar y a cada malla aproximada se le calcula la distancia generacional con respecto a $PF_{referencia}$.

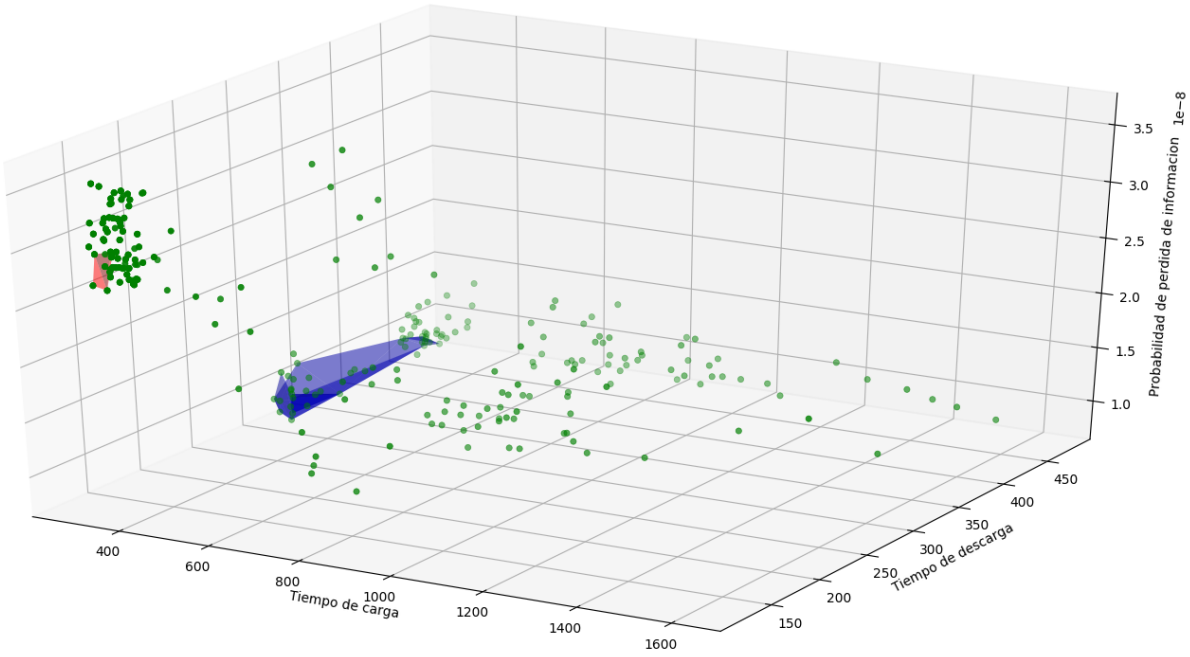

Figura 25. Malla de la configuración (3,5) y estrategia AdaptiveSecurity

En la Tabla 22, se muestra la distancia generacional de cada estrategia por configuración, cada elemento resaltado en negrita es el valor mínimo de columna, es decir, que configuración independientemente de la estrategia, se aproxima más a $PF_{referencia}$, en nuestro caso, la configuración (2,7) tiene el valor mínimo en dos de seis ocasiones, siendo la única configuración con más de un mínimo. Esto indica que la configuración (2,7) tiene el mejor comportamiento en general que el resto de las configuraciones.

Tabla 22. Distancia generacional por configuración.

(k,n)	AdaptiveSecurity	Random	BestUpload	BestDownload	AdaptiveSpeed	BestSecurity
(2,2)	0.060742	1.298537	0.781681	0.638712	0.107418	2.372624
(2,3)	0.015179	1.859058	**0.695746**	0.291060	0.020647	1.987032
(3,3)	1.208336	**1.093080**	1.194736	1.059468	1.198872	1.656761
(2,4)	0.003104	1.869200	1.167887	0.641964	0.004678	3.116897
(3,4)	1.438452	1.302600	1.211606	0.846982	1.179851	1.849106
(4,4)	0.946041	1.688644	1.415935	1.640427	1.769371	1.370296
(2,5)	0.000381	1.926968	0.967577	0.000894	0.000894	2.943359
(3,5)	0.210832	1.281921	0.729260	0.626846	0.626846	1.999627
(4,5)	1.280890	1.289199	1.550035	1.732216	1.732216	1.212554
(5,5)	1.356729	1.675953	1.163984	1.163984	1.163984	1.875286
(2,6)	**0.000087**	2.045615	1.048443	0.000185	0.795411	2.927188
(3,6)	1.358264	1.294212	0.892628	0.486175	0.879253	1.704536
(4,6)	1.757597	1.454496	1.328192	1.798185	1.333747	1.571807
(5,6)	1.213937	1.562278	1.188172	1.426618	1.188172	2.154903
(6,6)	0.952385	1.298505	1.744376	2.221205	1.742967	**1.155499**
(2,7)	2.454589	1.826612	1.546643	**0.000032**	**0.000032**	2.454589
(3,7)	1.935709	1.521193	1.160279	1.122760	1.122760	1.935709
(4,7)	1.526608	1.583557	1.765980	1.585160	1.585160	1.526608
(5,7)	1.510712	2.690261	1.449283	1.449283	1.449283	1.510712
(6,7)	1.176439	1.438115	1.909034	1.917591	1.917591	1.176439
(7,7)	1.182270	1.182270	1.182270	1.182270	1.182270	1.182270

En la Tabla 23, se muestra la distancia generacional de cada estrategia por configuración, cada elemento resaltado en negrita es el valor mínimo del renglón, es decir, la estrategia que se aproxima más a $PF_{referencia}$ para esa configuración. En nuestro caso, la estrategia AdaptiveSecurity tiene el valor mínimo en más del 50% ocasiones. Esto indica que la estrategia AdaptiveSecurity tiene mejor comportamiento en general que el resto de las estrategias.

Tabla 23. Distancia generacional por configuración.

(k,n)	AdaptiveSecurity	Random	BestUpload	BestDownload	AdaptiveSpeed	BestSecurity
(2,2)	**0.060742**	1.298537	0.781681	0.638712	0.107418	2.372624
(2,3)	**0.015179**	1.859058	0.695746	0.291060	0.020647	1.987032
(3,3)	1.208336	1.093080	1.194736	**1.059468**	1.198872	1.656761
(2,4)	**0.003104**	1.869200	1.167887	0.641964	0.004678	3.116897
(3,4)	1.438452	1.302600	1.211606	**0.846982**	1.179851	1.849106
(4,4)	**0.946041**	1.688644	1.415935	1.640427	1.769371	1.370296
(2,5)	**0.000381**	1.926968	0.967577	0.000894	0.000894	2.943359
(3,5)	**0.210832**	1.281921	0.729260	0.626846	0.626846	1.999627
(4,5)	**1.280890**	1.289199	1.550035	1.732216	1.732216	1.212554
(5,5)	1.356729	1.675953	**1.163984**	**1.163984**	**1.163984**	1.875286
(2,6)	**0.000087**	2.045615	1.048443	0.000185	0.795411	2.927188
(3,6)	1.358264	1.294212	0.892628	**0.486175**	0.879253	1.704536
(4,6)	1.757597	1.454496	**1.328192**	1.798185	1.333747	1.571807
(5,6)	1.213937	1.562278	**1.188172**	1.426618	**1.188172**	2.154903
(6,6)	**0.952385**	1.298505	1.744376	2.221205	1.742967	1.155499
(2,7)	2.454589	1.826612	1.546643	**0.000032**	**0.000032**	2.454589
(3,7)	1.935709	1.521193	**1.160279**	1.122760	1.122760	1.935709
(4,7)	**1.526608**	1.583557	1.765980	1.585160	1.585160	1.526608
(5,7)	1.510712	2.690261	**1.449283**	**1.449283**	**1.449283**	1.510712
(6,7)	**1.176439**	1.438115	1.909034	1.917591	1.917591	1.176439
(7,7)	1.182270	1.182270	1.182270	1.182270	1.182270	1.182270

Capítulo 5. Conclusiones y trabajo futuro

Para minimizar el tiempo de acceso a los datos y la probabilidad de pérdida de información, propusimos e implementamos un modelo adaptativo de almacenamiento de datos basado en esquemas de compartición de secretos y el sistema numérico de residuo en un entorno multi-nube.

Para esto, hicimos uso de un API universal que unifica la administración de datos y realizamos mediciones de los tiempos de transferencia entre cada una de las nubes y los tiempos de carga y descarga. Esto reveló que el ancho de banda para cargar y descargar archivos por medio del API es limitado por lo que se implementó el API brindado de cada proveedor de servicios.

Analizamos información reportada sobre la cantidad de interrupciones de servicio no planeadas durante el año 2015 de distintos proveedores de servicio en la nube para determinar probabilidades de falla de las nubes (negación de acceso) que se aproximen a la realidad.

Propusimos seis estrategias de asignación de datos para seleccionar los mejores proveedores de servicio en la nube basados en sus velocidades de acceso y probabilidad de falla. Evaluamos el comportamiento de rendimiento de estrategias en siete proveedores de servicio en la nube bien conocidos y diferentes parámetros del sistema.

Presentamos evidencia experimental de que el uso de estrategias adaptativas mejora el rendimiento del sistema. AdaptiveSpeed ofrece mejores resultados para la velocidad de carga y descarga, aunque a expensas de la probabilidad de pérdida de información. Mientras tanto, AdaptiveSecurity mejorará la velocidad de carga y la probabilidad de pérdida de información, a expensas de la velocidad de descarga. Las soluciones de cada estrategia no se encuentran esparcidas por todo el espacio de soluciones, si no se conglomeran en espacios similares, esto podría indicar que los cambios en los parámetros de velocidad y probabilidad de falla de las nubes no afectan el rendimiento de las estrategias.

Realizamos un estudio en el dominio tridimensional, para comparar el comportamiento de las estrategias y configuraciones de una forma más concreta, ya que el estudio en dos dimensiones suele proporcionar resultados incompletos. Para esto, hicimos uso de la distancia generacional cómo métrica de comparación y encontramos que la configuración (2,7) tiene un mejor rendimiento en general, al igual que la estrategia AdaptiveSecurity, la cual presentó una distancia más cercana a $PF_{referencia}$ en 11 de 21 configuraciones.

Durante la elaboración de este trabajo, encontramos varios problemas al momento de caracterizar las nubes. Cuándo realizamos los experimentos para medir las velocidades de las nubes con el API de Kloudless, nos encontramos en un estado de confusión al desconocer el motivo por el cual los resultados no mantenían congruencia con trabajos realizados con anterioridad. Esto debido a que desconocíamos y desconocemos hasta la fecha, la ubicación física de cada una de las nubes y por lo tanto la distancia entre ellas, el ancho de banda que utilizan, etc. Para resolver esa incertidumbre, es posible rentar infraestructura como servicio de los distintos proveedores y elegir los valores de esas variables que desconocemos, además se puede rentar servicio de procesamiento para aprovechar todas las propiedades que brinda RRNS.

Para poder determinar la probabilidad de falla de cada una de las nubes, la tarea fue complicada. La mayoría de los trabajos publicados hacen una recopilación manual por un cierto periodo de tiempo de los reportes presentados por los proveedores de servicio, en nuestro caso, una tarea de esa índole no nos sería fructífera, ya que tendríamos que determinar cada cuánto hacer la recopilación manual para actualizar los valores de las nubes. Además, no hay certeza de que los proveedores reporten cada una de sus fallas. Para esto, el utilizar herramientas como Panopta para monitorear infraestructura rentada, brindaría información actualizada de la probabilidad de falla de las nubes rentadas.

Como trabajo futuro, se propone el uso de estrategias más sofisticadas que exploren y exploten el espacio de solución, tomando en cuenta los tres criterios de optimización.

Literatura citada

AlZain, M. A., Pardede, E., Soh, B., & Thom, J. A. (2011). Cloud computing security: From single to multi-clouds. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 5490–5499. https://doi.org/10.1109/HICSS.2012.153

Asmuth, C., & Bloom, J. (1983). A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, *29*(2), 208–210. https://doi.org/10.1109/TIT.1983.1056651

Babitha M.P., & Babu, K. R. R. (2016). Secure cloud storage using AES encryption. *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, 859–864. https://doi.org/10.1109/ICACDOT.2016.7877709

Barsi, F., & Maestrini, P. (1973). Error Correcting Properties of Redundant Residue Number Systems. *IEEE Transactions on Computers*, *C*(3), 307–315.

Basescu, C., Cachin, C., Haas, R., & Antipolis, F.-S. (2011). Brief Announcement : Robust Data Sharing with Key-Value Stores. *Distributed Computing*, *3802*, 221–222.

Bessani, A., Correia, M., Quaresma, B., Sousa, P., André, F., & Sousa, P. (2013). DepSky. *Proceedings of the Sixth Conference on Computer Systems - EuroSys '11*, *00*(00), 31. https://doi.org/10.1145/1966445.1966449

Blakley, G. R. (1979). Safeguarding cryptographic keys. *Afips*, 313. https://doi.org/10.1109/AFIPS.1979.98

Butler, B. (2015). And the cloud provider with the best uptime in 2015 is…. Retrieved from https://www.cio.com/article/3020374/cloud-computing/and-the-cloud-provider-with-the-best-uptime-in-2015-is.html

Celesti, A., Fazio, M., Villari, M., & Puliafito, A. (2016). Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems. *Journal of Network and Computer Applications*, *59*, 208–218. https://doi.org/10.1016/j.jnca.2014.09.021

Chervyakov, N., Babenko, M., Tchernykh, A., Kucherov, N., Miranda-López, V., & Cortés-Mendoza, J. M. (2017). AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2017.09.061

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, Third Edition* (3rd ed.). The MIT Press.

Daemen, J., & Rijmen, V. (2003). *The Rijndael Block Cipher: AES Proposal*. *Nist*. Retrieved from http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf

Deb, K., & Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc.

Ermakova, T., & Fabian, B. (2013). Secret sharing for health data in multi-provider clouds. *Proceedings - 2013 IEEE International Conference on Business Informatics, IEEE CBI 2013*, (May 2014), 93–100. https://doi.org/10.1109/CBI.2013.22

Flores, I. (1969). Residue Arithmetic and Its Application to Computer Technology (Nicholas S. Szabo and Richard I. Tanaka). *SIAM Review*, *11*(1), 103–104. https://doi.org/10.1137/1011027

Gasser, M. (1988). *Building a Secure Computer System*. New York: Van Nostrand Reinhold.

Lauer, M. (2011). Data Security in the Cloud Why Cloud Computing ? *Seminar*, 61–64. https://doi.org/10.1109/MSP.2009.87

Lin, S.-J., Chung, W.-H., & Han, Y. S. (2014). Novel Polynomial Basis and Its Application to Reed-Solomon Erasure Codes. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science* (pp. 316–325). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/FOCS.2014.41

Luc, D. T. (2016). *Multiobjective Linear Programming*. *Multiobjective Linear Programming: An Introduction*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-21091-9

Marium, S., Nazir, Q., Ahmed Shaikh, A., Ahthasham, S., & Aamir Mehmood, M. (2012). Implementation of Eap with RSA for Enhancing The Security of Cloud Computing. *International Journal of Basic and Applied Sciences*, *1*.

Mell, P. M., & Grance, T. (2011). The NIST definition of cloud computing. https://doi.org/10.6028/NIST.SP.800-145

Mignotte, M. (1983). How To Share a Secret. *Communications of the ACM (CACM)*, *22*(1), 612–613. https://doi.org/http://doi.acm.org/10.1145/359168.359176

Miranda-López, V., Tchernykh, A., Cortés-Mendoza, J. M., Babenko, M., Radchenko, G., Nesmachnow, S., & Du, Z. (2018). Experimental Analysis of Secret Sharing Schemes for Cloud Storage Based on RNS. In E. Mocskos & S. Nesmachnow (Eds.), *High Performance Computing* (pp. 370–383). Cham: Springer International Publishing.

Morgan, L. (2018). List of data breaches and cyber attacks in May 2018. Retrieved from https://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-may-2018-17273571-records-leaked/

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04101-3

Pundkar, S. N., & Shekokar, N. (2016). Cloud computing security in multi-clouds using Shamir's secret sharing scheme. *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, 392–395. https://doi.org/10.1109/ICEEOT.2016.7755427

Rabin, M. O. (1989). Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, *36*(2), 335–348. https://doi.org/10.1145/62044.62050

Rathanam, G. J., & Sumalatha, M. R. (2014). Dynamic secure storage system in cloud services. *2014 International Conference on Recent Trends in Information Technology, ICRTIT 2014*. https://doi.org/10.1109/ICRTIT.2014.6996175

Reed, I. S., & Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 300–304. https://doi.org/https://doi.org/10.1137/0108018

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120–126. https://doi.org/10.1145/359340.359342

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, *22*(11), 612–613. https://doi.org/10.1145/359168.359176

Tchernykh, A., Babenko, M., Chervyakov, N., Miranda-López, V., Kuchukov, V., Cortés-Mendoza, J. M., … Avetisyan, A. (2018). AC-RRNS: Anti-collusion secured data sharing scheme for cloud storage. *International Journal of Approximate Reasoning*, *102*, 60–73. https://doi.org/10.1016/j.ijar.2018.07.010

Tchernykh, A., Schwiegelsohn, U., Talbi, E. ghazali, & Babenko, M. (2016). Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. *Journal of Computational Science*. https://doi.org/10.1016/j.jocs.2016.11.011

Tos, U., Mokadem, R., Hameurlain, A., Ayav, T., & Bora, S. (2015). Dynamic replication strategies in data grid systems: a survey. *Journal of Supercomputing*, *71*(11), 4116–4140. https://doi.org/10.1007/s11227-015-1508-7

Van Veldhuizen, D. A., & Lamont, G. B. (2000). On Measuring Multiobjective Evolutionary Algorithm Performance. *Proceedings of the IEEE Congress on Evolutionary Computation*, *1*, 204–211. https://doi.org/10.1109/CEC.2000.870296

Anexos

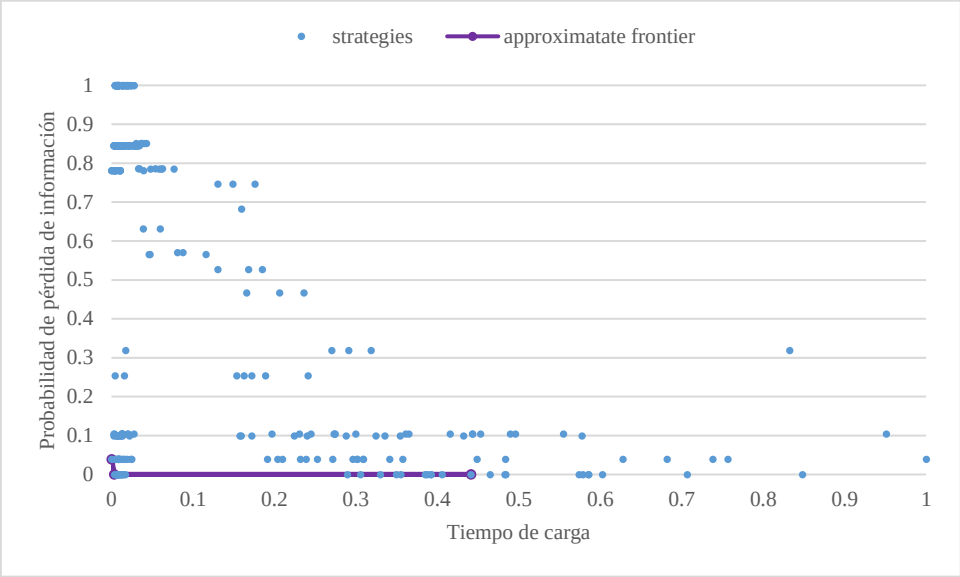

Figura 26. Aproximación de Pareto para la configuración $(2,2)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 24. Miembros del frente aproximado de Pareto de la configuración $(2,2)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	909	546	0.001519	0.441259	0	0.955056
AdaptiveSecurity	130	560	0.001519	0.003373	5.26E-08	0.986517
AdaptiveSecurity	124	361	0.001605	0	0.038699	0.539326

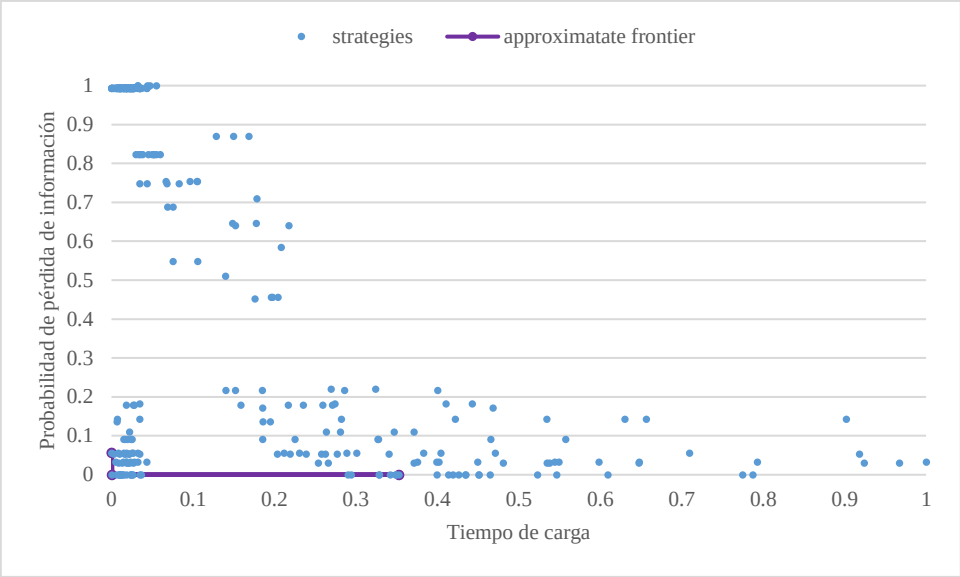

Figura 27. Aproximación de Pareto para la configuración $(2,3)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 25. Miembros del frente aproximado de Pareto de la configuración $(2,3)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	884	379	2.36E-06	0.352851	0	0.579775
AdaptiveSecurity	198	361	2.36E-06	0.000514	6.43E-17	0.539326
AdaptiveSecurity	197	357	2.73E-06	0	0.056062	0.530337

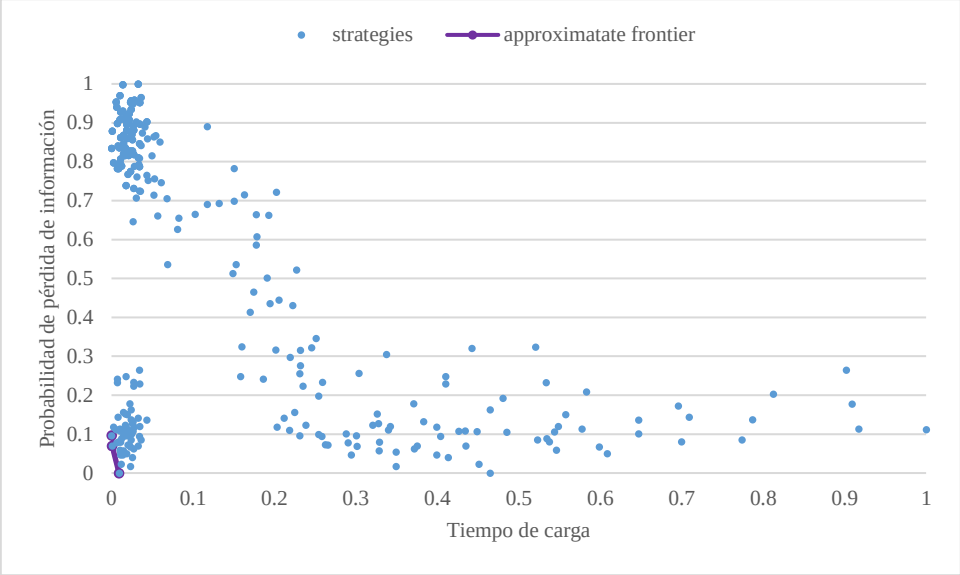

Figura 28. Aproximación de Pareto para la configuración $(3,3)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 26. Miembros del frente aproximado de Pareto de la configuración $(3,3)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	143	458	0.002359	0.009266	0	0.954286
AdaptiveSecurity	132	431	0.002549	0.000772	0.069849	0.877143
AdaptiveSecurity	131	339	0.002622	0	0.096475	0.614286

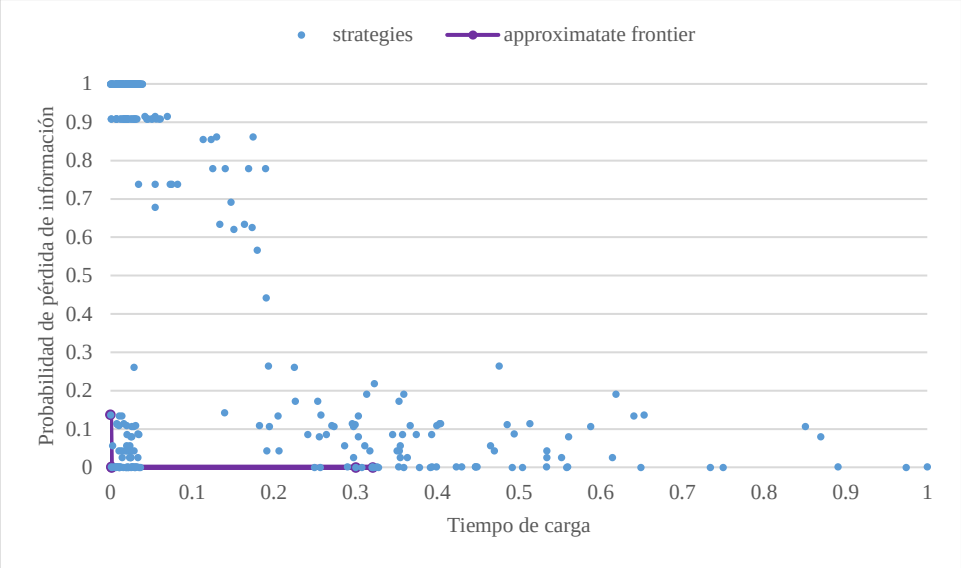

Figura 29. Aproximación de Pareto para la configuración $(2,4)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 27. Miembros del frente aproximado de Pareto de la configuración $(2,4)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	971	358	3.89E-09	0.321006	0	0.532584
Random	929	342	3.89E-09	0.300296	2.88E-09	0.496629
AdaptiveSecurity	323	354	3.89E-09	0.001479	2.22E-08	0.523596
AdaptiveSecurity	322	361	3.92E-09	0.000986	0.001893	0.539326
AdaptiveSecurity	320	384	5.84E-09	0	0.137309	0.591011

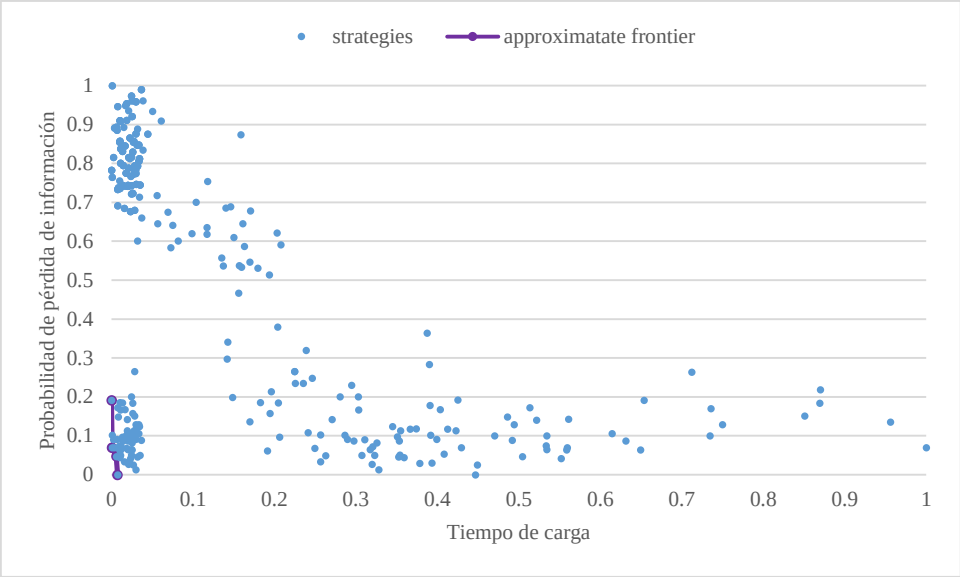

Figura 30. Aproximación de Pareto para la configuración $(3,4)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 28. Miembros del frente aproximado de Pareto de la configuración $(3,4)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	223	458	4.58E-06	0.007418	0	0.954286
AdaptiveSecurity	221	429	5.1E-06	0.005935	0.047339	0.871429
AdaptiveSecurity	214	431	5.35E-06	0.000742	0.069791	0.877143
AdaptiveSecurity	213	316	6.69E-06	0	0.191365	0.548571

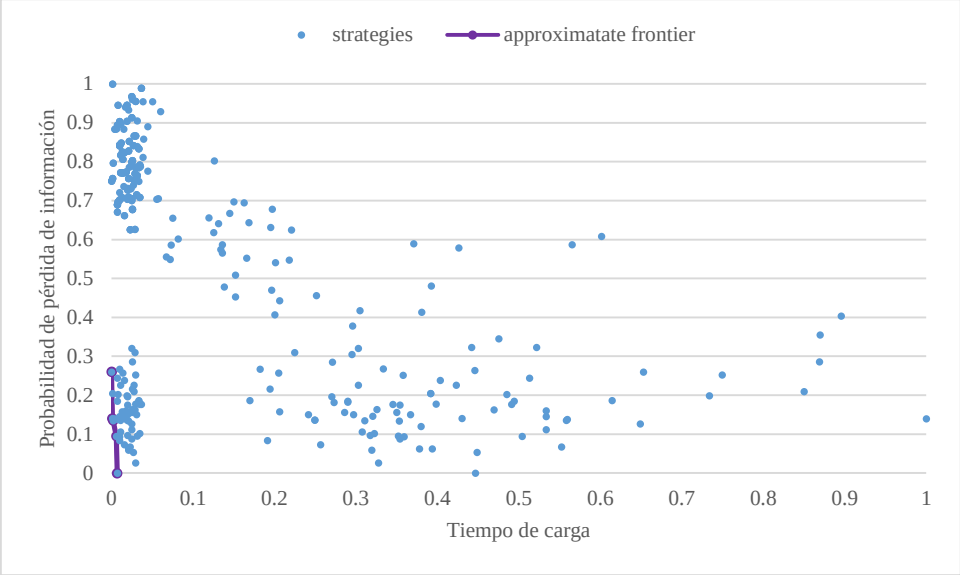

Figura 31. Aproximación de Pareto para la configuración $(4,4)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 29. Miembros del frente aproximado de Pareto de la configuración $(4,4)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	167	389	0.003258	0.006903	0	0.930147
AdaptiveSecurity	166	355	0.003553	0.005917	0.094837	0.805147
AdaptiveSecurity	162	353	0.003678	0.001972	0.135171	0.797794
AdaptiveSecurity	161	367	0.003695	0.000986	0.140591	0.849265
AdaptiveSecurity	160	268	0.004067	0	0.260122	0.485294

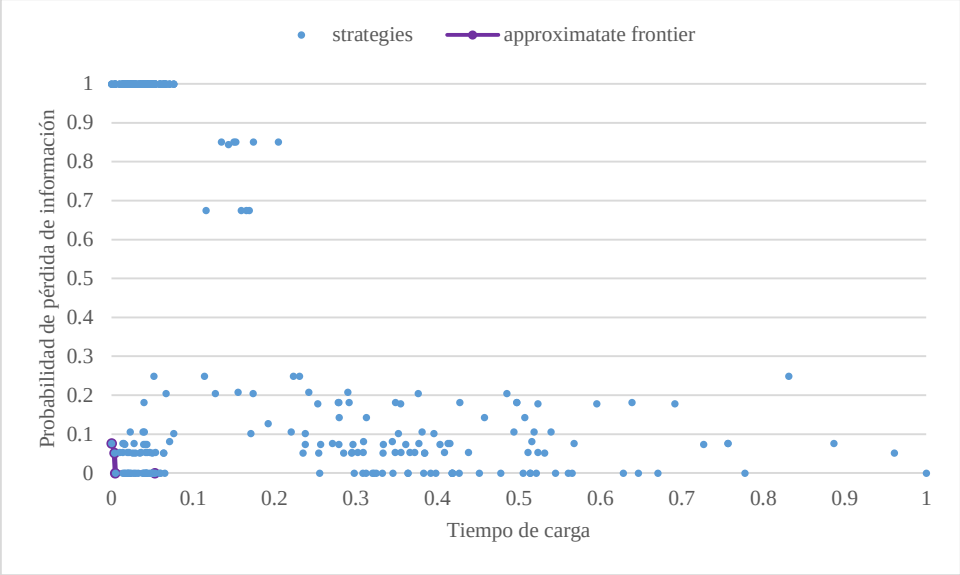

Figura 32. Aproximación de Pareto para la configuración $(2,5)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 30. Miembros del frente aproximado de Pareto de la configuración $(2,5)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado	Probabilidad de falla normalizada
AdaptiveSecurity	555	353	6.38E-12	0.053408	0	0.521348	0
AdaptiveSecurity	459	560	6.38E-12	0.004578	2.78E-09	0.986517	2.78E-09
AdaptiveSecurity	457	548	7.67E-12	0.003561	0.051792	0.959551	0.051792
AdaptiveSecurity	450	384	8.29E-12	0	0.076506	0.591011	0.076506

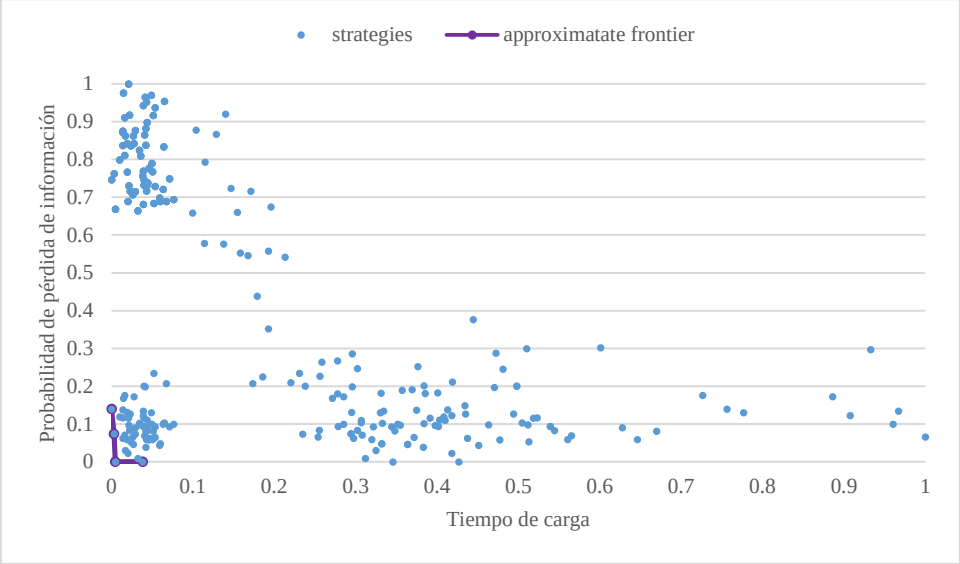

Figura 33. Aproximación de Pareto para la configuración $(3,5)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 31. Miembros del frente aproximado de Pareto de la configuración $(3,5)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	349	287	8.37478E-09	0.038226	0	0.465714
AdaptiveSecurity	305	458	8.37933E-09	0.004587	0.000166	0.954286
AdaptiveSecurity	303	421	1.04004E-08	0.003058	0.073982	0.848571
AdaptiveSecurity	299	316	1.21972E-08	0	0.139608	0.548571

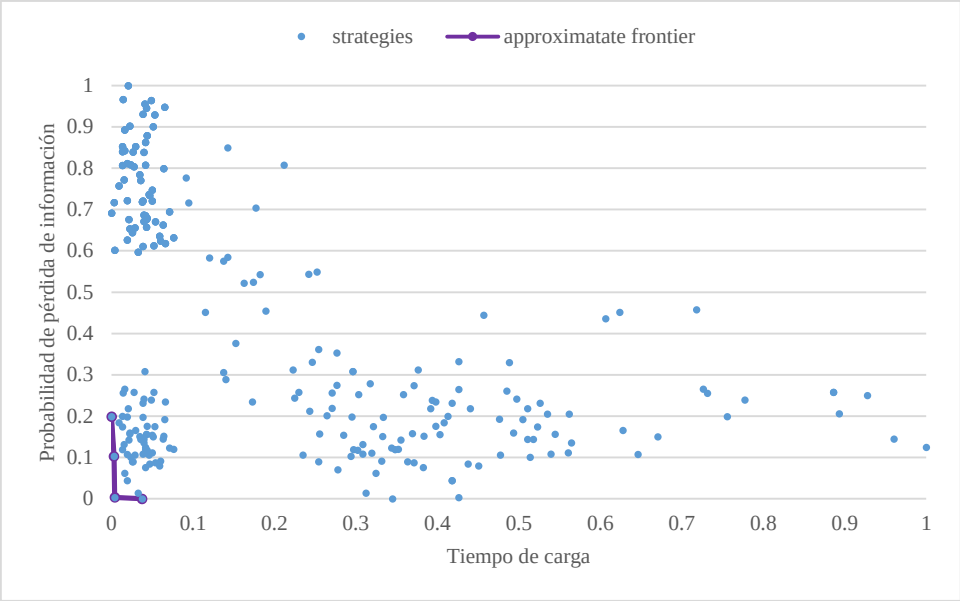

Figura 34. Aproximación de Pareto para la configuración $(4,5)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 32. Miembros del frente aproximado de Pareto de la configuración $(4,5)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	262	356	7.8E-06	0.03764	0	0.808824
AdaptiveSecurity	229	389	7.85E-06	0.004069	0.003529	0.930147
AdaptiveSecurity	228	353	9.33E-06	0.003052	0.103035	0.797794
AdaptiveSecurity	225	268	1.08E-05	0	0.198934	0.485294

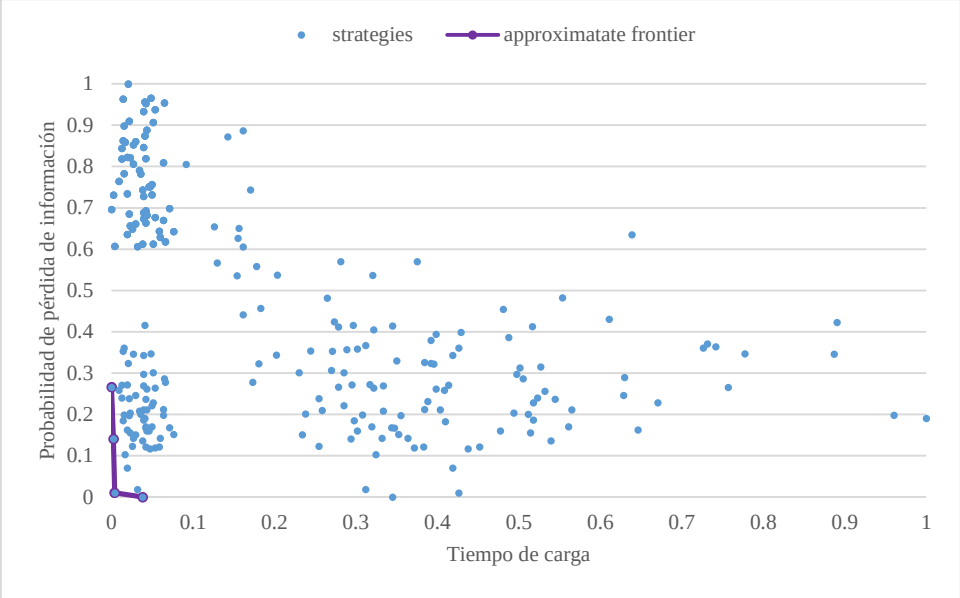

Figura 35. Aproximación de Pareto para la configuración $(5,5)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 33. Miembros del frente aproximado de Pareto de la configuración $(5,5)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	210	319	0.004232	0.038168	0	0.832536
AdaptiveSecurity	183	337	0.004265	0.003817	0.010343	0.91866
AdaptiveSecurity	182	308	0.004685	0.002545	0.141066	0.779904
AdaptiveSecurity	180	329	0.005086	0	0.266035	0.880383

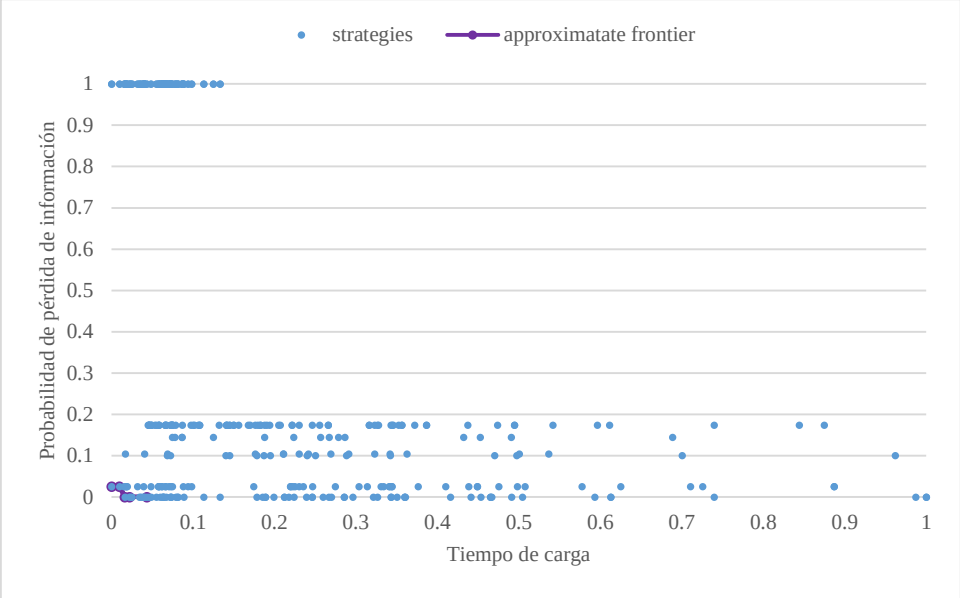

Figura 36. Aproximación de Pareto para la configuración $(2,6)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 34. Miembros del frente aproximado de Pareto de la configuración $(2,6)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	818	560	1.24E-14	0.043206	0	0.986517
AdaptiveSecurity	781	334	1.24E-14	0.022172	4.56E-08	0.478652
AdaptiveSecurity	771	558	1.24E-14	0.016487	4.56E-08	0.982022
AdaptiveSecurity	759	554	1.34E-14	0.009665	0.025429	0.973034
AdaptiveSecurity	742	384	1.34E-14	0	0.025429	0.591011

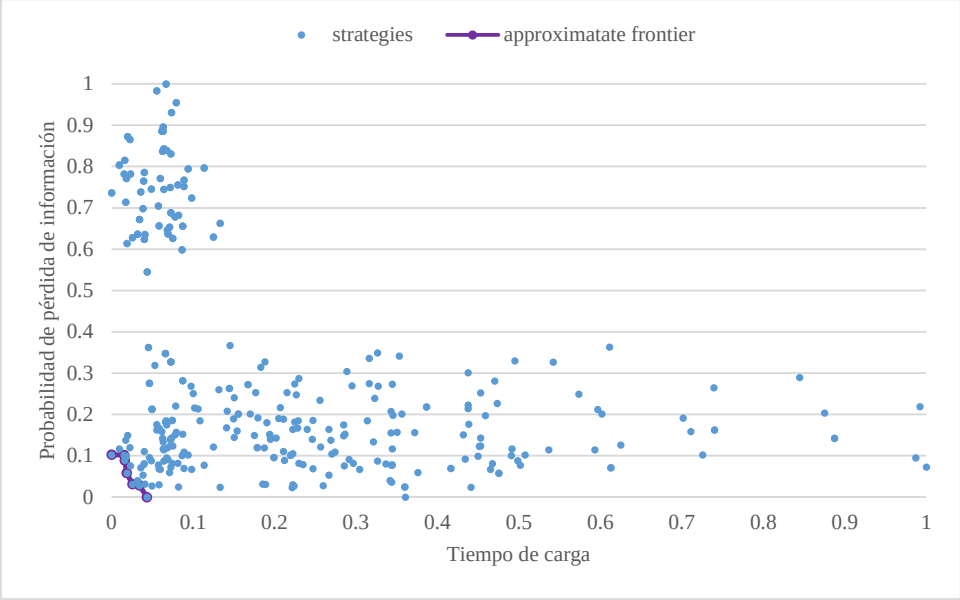

Figura 37. Aproximación de Pareto para la configuración $(3,6)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 35. Miembros del frente aproximado de Pareto de la configuración $(3,6)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	544	458	1.69E-11	0.04359	0	0.954286
AdaptiveSecurity	533	287	1.84E-11	0.034188	0.02837	0.465714
AdaptiveSecurity	523	410	1.86E-11	0.025641	0.03119	0.817143
AdaptiveSecurity	515	409	2E-11	0.018803	0.05864	0.814286
AdaptiveSecurity	512	435	2.16E-11	0.016239	0.088886	0.888571
AdaptiveSecurity	511	283	2.23E-11	0.015385	0.101752	0.454286
AdaptiveSecurity	493	316	2.23E-11	0	0.102483	0.548571

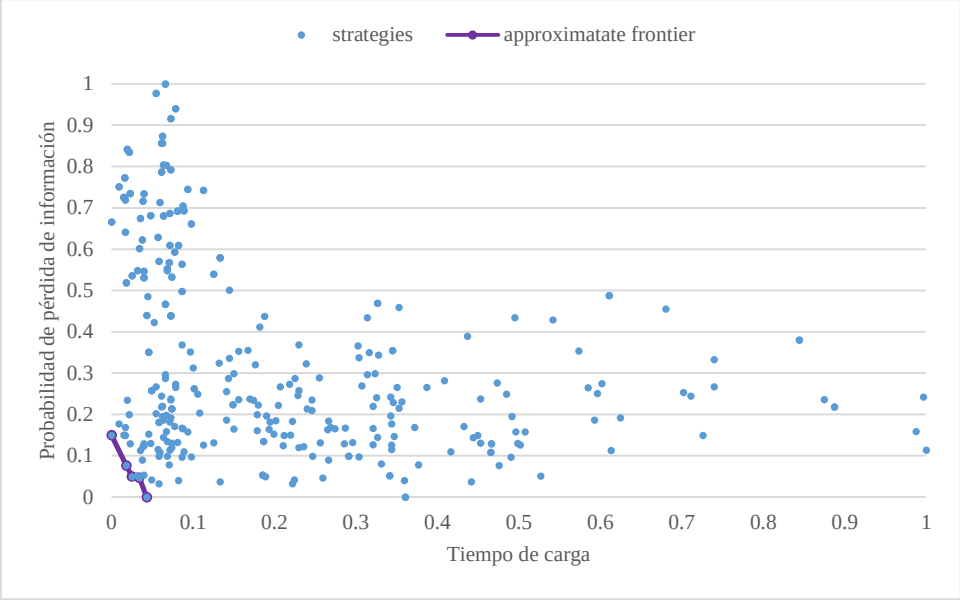

Figura 38. Aproximación de Pareto para la configuración $(4,6)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 36. Miembros del frente aproximado de Pareto de la configuración $(4,6)$ para probabilidad de pérdida de información vs tiempo de carga

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	409	389	1.84E-08	0.043231	0	0.930147
AdaptiveSecurity	401	356	2.04E-08	0.03413	0.046809	0.808824
AdaptiveSecurity	393	353	2.05E-08	0.025028	0.049934	0.797794
AdaptiveSecurity	387	338	2.16E-08	0.018203	0.076744	0.742647
AdaptiveSecurity	371	268	2.47E-08	0	0.149723	0.485294

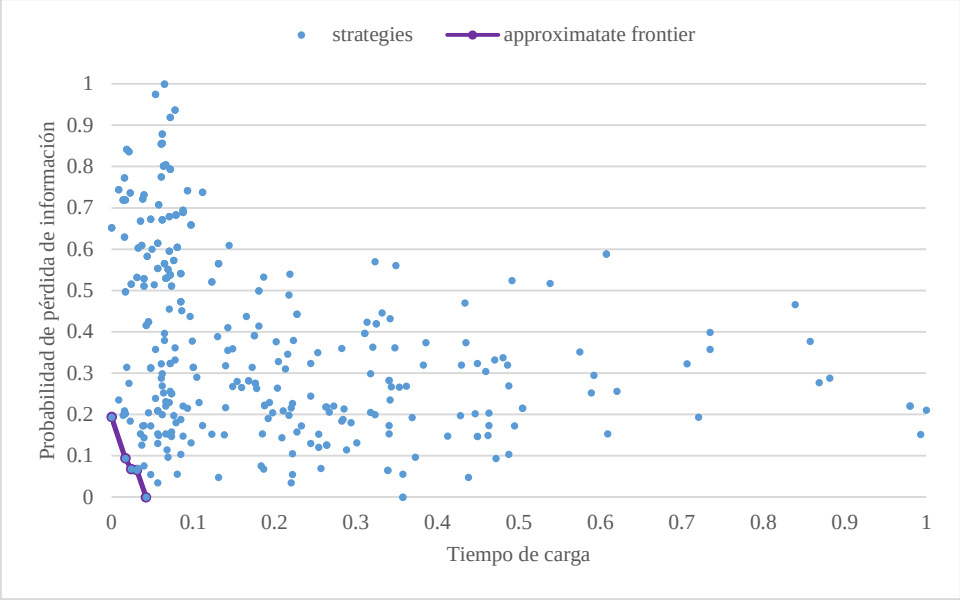

Figura 39. Aproximación de Pareto para la configuración $(5,6)$. Probabilidad de pérdida de información vs tiempo de carga

Tabla 37. Miembros del frente aproximado de Pareto de la configuración $(5,6)$ para probabilidad de pérdida de información vs tiempo de carga

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	327	337	1.33E-05	0.042373	0	0.91866
AdaptiveSecurity	319	351	1.44E-05	0.031073	0.06537	0.985646
AdaptiveSecurity	314	337	1.45E-05	0.024011	0.068045	0.91866
AdaptiveSecurity	309	305	1.49E-05	0.016949	0.094202	0.76555
AdaptiveSecurity	297	329	1.66E-05	0	0.194162	0.880383

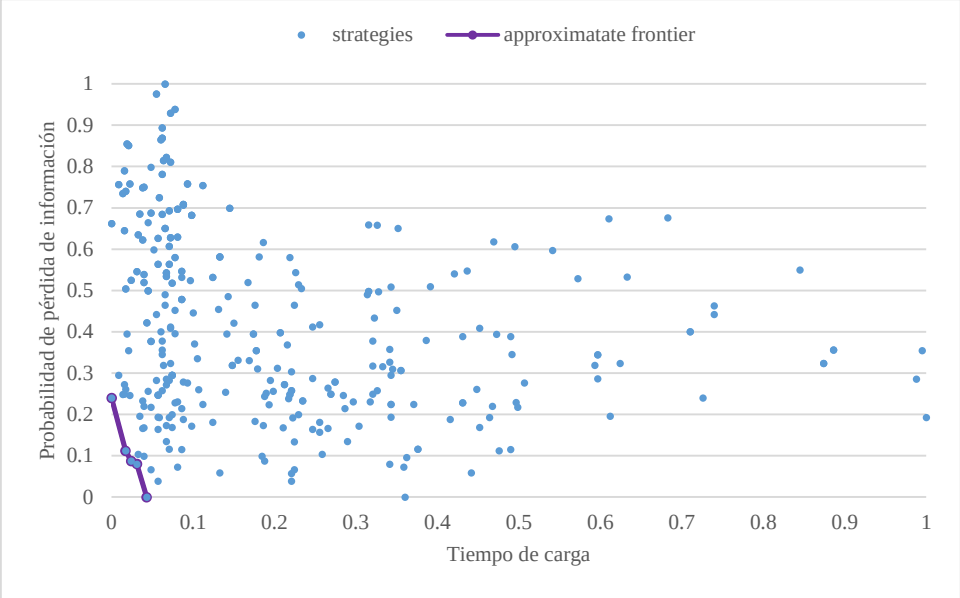

Figura 40. Aproximación de Pareto para la configuración $(6,6)$. Probabilidad de pérdida de información vs tiempo de carga

Tabla 38. Miembros del frente aproximado de Pareto de la configuración $(6,6)$ para probabilidad de pérdida de información vs tiempo de carga

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
AdaptiveSecurity	270	299	0.005514	0.043103	0	0.842593
AdaptiveSecurity	263	310	0.005754	0.031034	0.080153	0.944444
AdaptiveSecurity	259	307	0.005775	0.024138	0.087381	0.916667
AdaptiveSecurity	255	290	0.00585	0.017241	0.112343	0.759259
AdaptiveSecurity	245	292	0.006232	0	0.240169	0.777778

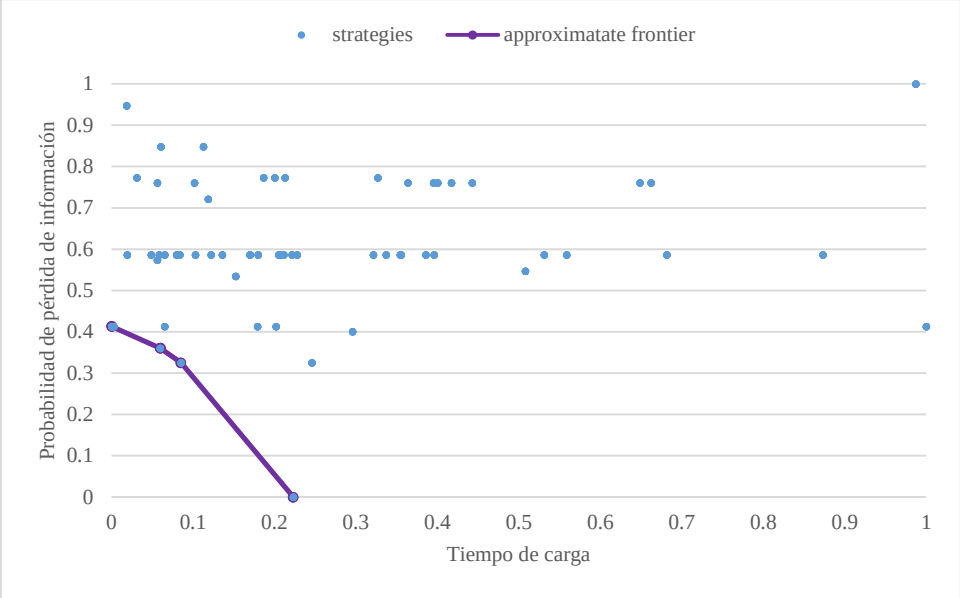

Figura 41. Aproximación de Pareto para la configuración $(2,7)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 39. Miembros del frente aproximado de Pareto de la configuración (2,7) para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	1443	560	2.59E-17	0.223061	0	0.986517
AdaptiveSecurity	1443	560	2.59E-17	0.223061	0	0.986517
BestSecurity	1242	353	2.59E-17	0.085106	0.325183	0.521348
AdaptiveSecurity	1242	353	2.59E-17	0.085106	0.325183	0.521348
Random	1205	353	2.59E-17	0.059712	0.360536	0.521348
BestUpload	1205	157	2.59E-17	0.059712	0.360536	0.080899
BestDownload	1205	123	2.59E-17	0.059712	0.360536	0.004494
AdaptiveSpeed	1205	157	2.59E-17	0.059712	0.360536	0.080899
Random	1118	548	2.59E-17	0	0.413064	0.959551
BestSecurity	1118	548	2.59E-17	0	0.413064	0.959551
AdaptiveSecurity	1118	548	2.59E-17	0	0.413064	0.959551

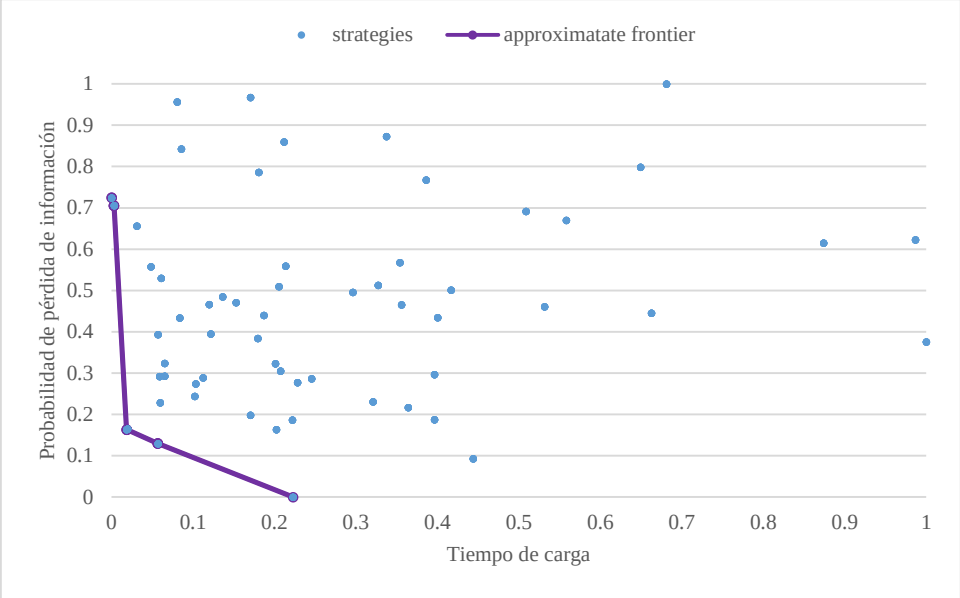

Figura 42. Aproximación de Pareto para la configuración (3,7). Probabilidad de pérdida de información vs tiempo de carga.

Tabla 40. Miembros del frente aproximado de Pareto de la configuración $(3,7)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	959	416	3.5E-14	0.22291	0	0.834286
BestUpload	798	126	3.81E-14	0.05676	0.129803	0.005714
BestDownload	798	126	3.81E-14	0.05676	0.129803	0.005714
AdaptiveSpeed	798	126	3.81E-14	0.05676	0.129803	0.005714
Random	761	280	3.89E-14	0.018576	0.163328	0.445714
BestSecurity	761	430	3.89E-14	0.018576	0.163328	0.874286
AdaptiveSecurity	761	430	3.89E-14	0.018576	0.163328	0.874286
Random	746	312	5.18E-14	0.003096	0.70555	0.537143
BestUpload	746	149	5.18E-14	0.003096	0.70555	0.071429
AdaptiveSpeed	746	149	5.18E-14	0.003096	0.70555	0.071429
Random	743	278	5.22E-14	0	0.724693	0.44
BestDownload	743	134	5.22E-14	0	0.724693	0.028571

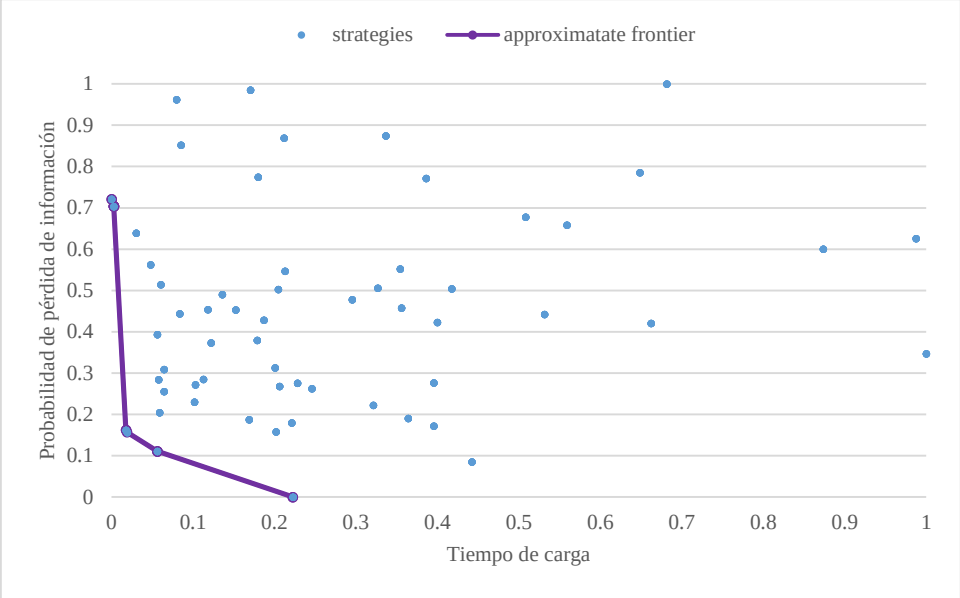

Figura 43. Aproximación de Pareto para la configuración $(4,7)$. Probabilidad de pérdida de información vs tiempo de carga.

Tabla 41. Miembros del frente aproximado de Pareto de la configuración $(4,7)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestUpload	721	161	4.17E-11	0.222527	0	0.091912
AdaptiveSpeed	721	161	4.17E-11	0.222527	0	0.091912
BestUpload	600	137	4.59E-11	0.056319	0.110898	0.003676
BestDownload	600	137	4.59E-11	0.056319	0.110898	0.003676
AdaptiveSpeed	600	137	4.59E-11	0.056319	0.110898	0.003676
Random	573	340	4.76E-11	0.019231	0.156704	0.75
BestSecurity	572	354	4.78E-11	0.017857	0.162156	0.801471
AdaptiveSecurity	572	354	4.78E-11	0.017857	0.162156	0.801471
Random	561	259	6.83E-11	0.002747	0.703694	0.452206
BestUpload	561	158	6.83E-11	0.002747	0.703694	0.080882
BestDownload	561	151	6.83E-11	0.002747	0.703694	0.055147
AdaptiveSpeed	561	158	6.83E-11	0.002747	0.703694	0.080882
BestSecurity	561	350	6.83E-11	0.002747	0.703694	0.786765
AdaptiveSecurity	561	350	6.83E-11	0.002747	0.703694	0.786765
BestSecurity	559	391	6.89E-11	0	0.721198	0.9375
AdaptiveSecurity	559	391	6.89E-11	0	0.721198	0.9375

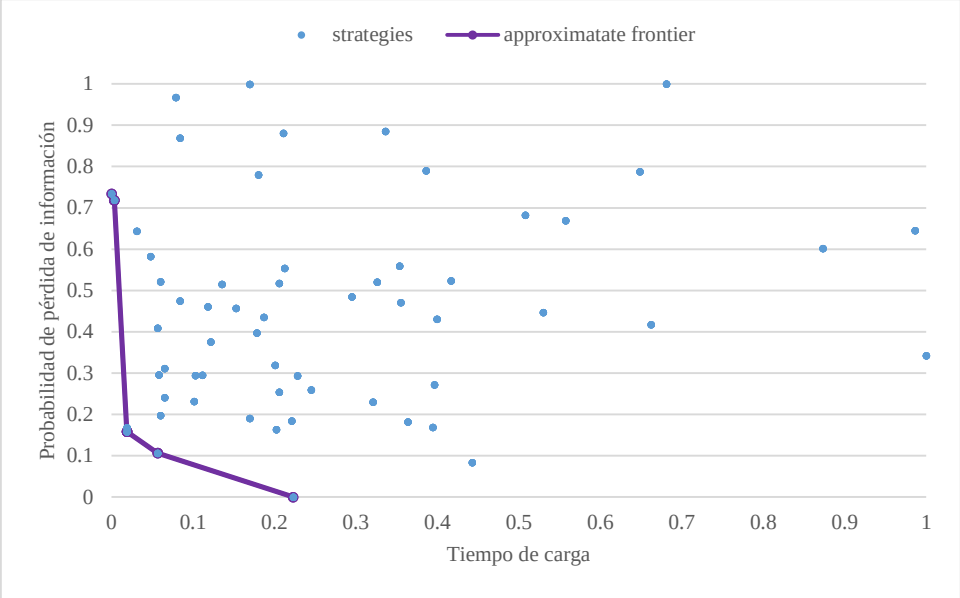

Figura 44. Aproximación de Pareto para la configuración (5,7). Probabilidad de pérdida de información vs tiempo de carga.

Tabla 42. Miembros del frente aproximado de Pareto de la configuración $(5,7)$ para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	577	337	3.61E-08	0.222985	0	0.91866
AdaptiveSecurity	577	337	3.61E-08	0.222985	0	0.91866
BestUpload	480	171	3.92E-08	0.056604	0.106565	0.124402
BestDownload	480	171	3.92E-08	0.056604	0.106565	0.124402
AdaptiveSpeed	480	171	3.92E-08	0.056604	0.106565	0.124402
Random	458	255	4.07E-08	0.018868	0.158406	0.526316
BestUpload	458	176	4.07E-08	0.018868	0.158406	0.148325
BestDownload	458	176	4.07E-08	0.018868	0.158406	0.148325
AdaptiveSpeed	458	176	4.07E-08	0.018868	0.158406	0.148325
BestSecurity	458	337	4.07E-08	0.018868	0.158406	0.91866
AdaptiveSecurity	458	337	4.07E-08	0.018868	0.158406	0.91866
Random	449	319	5.7E-08	0.003431	0.718822	0.832536
BestUpload	449	160	5.7E-08	0.003431	0.718822	0.07177
AdaptiveSpeed	449	160	5.7E-08	0.003431	0.718822	0.07177
BestSecurity	449	316	5.7E-08	0.003431	0.718822	0.818182
AdaptiveSecurity	449	316	5.7E-08	0.003431	0.718822	0.818182
BestSecurity	447	337	5.75E-08	0	0.734331	0.91866
AdaptiveSecurity	447	337	5.75E-08	0	0.734331	0.91866

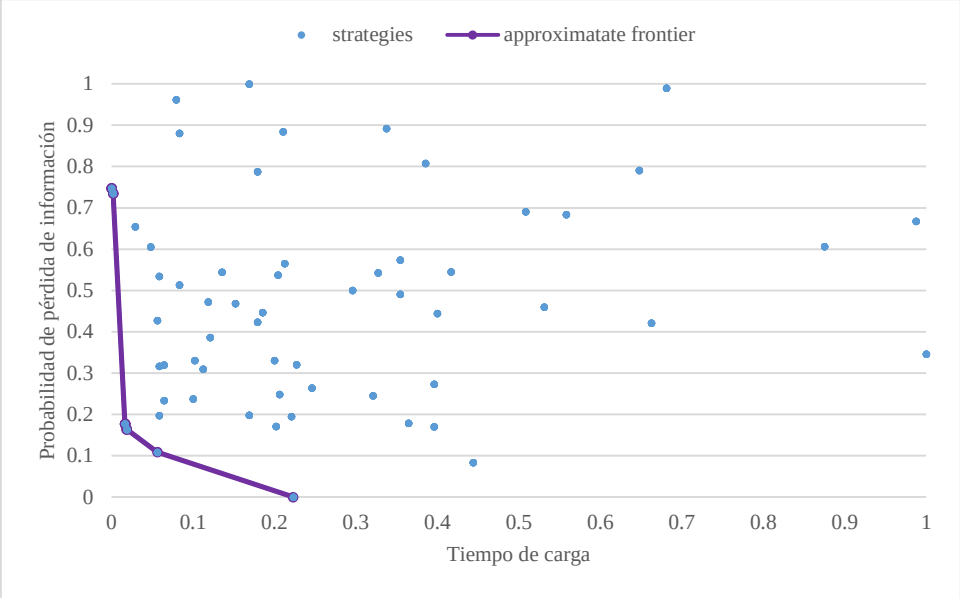

Figura 45. Aproximación de Pareto para la configuración (6,7). Probabilidad de pérdida de información vs tiempo de carga.

Tabla 43. Miembros del frente aproximado de Pareto de la configuración (6,7) para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	476	291	2.05E-05	0.222917	0	0.768519
Random	396	307	2.17E-05	0.05625	0.108856	0.916667
Random	378	307	2.24E-05	0.01875	0.163859	0.916667
BestSecurity	378	307	2.24E-05	0.01875	0.163859	0.916667
AdaptiveSecurity	378	307	2.24E-05	0.01875	0.163859	0.916667
Random	377	284	2.25E-05	0.016667	0.177329	0.703704
BestUpload	377	220	2.25E-05	0.016667	0.177329	0.111111
AdaptiveSpeed	377	220	2.25E-05	0.016667	0.177329	0.111111
BestSecurity	377	289	2.25E-05	0.016667	0.177329	0.75
AdaptiveSecurity	377	289	2.25E-05	0.016667	0.177329	0.75
BestSecurity	370	293	2.9E-05	0.002083	0.734937	0.787037
AdaptiveSecurity	370	293	2.9E-05	0.002083	0.734937	0.787037
Random	369	232	2.92E-05	0	0.747333	0.222222
BestDownload	369	232	2.92E-05	0	0.747333	0.222222
BestSecurity	369	298	2.92E-05	0	0.747333	0.833333
AdaptiveSecurity	369	298	2.92E-05	0	0.747333	0.833333

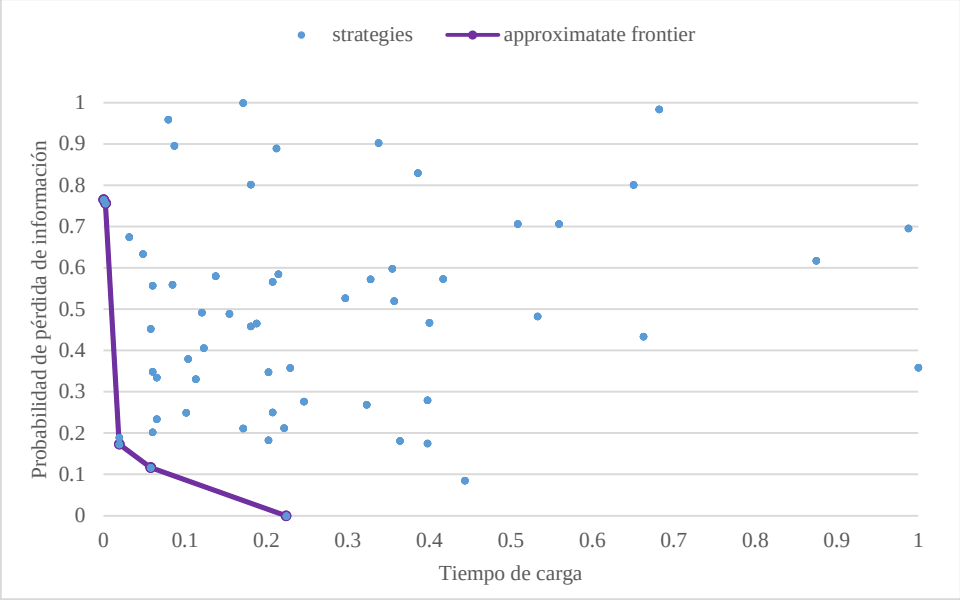

Figura 46. Aproximación de Pareto para la configuración (7,7). Probabilidad de pérdida de información vs tiempo de carga.

Tabla 44. Miembros del frente aproximado de Pareto de la configuración (7,7) para probabilidad de pérdida de información vs tiempo de carga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	411	277	0.00682	0.224096	0	0.5625
BestDownload	411	277	0.00682	0.224096	0	0.5625
BestUpload	342	283	0.007037	0.057831	0.116945	0.75
BestDownload	342	283	0.007037	0.057831	0.116945	0.75
AdaptiveSpeed	342	283	0.007037	0.057831	0.116945	0.75
BestSecurity	326	283	0.007142	0.019277	0.173255	0.75
AdaptiveSecurity	326	283	0.007142	0.019277	0.173255	0.75
BestUpload	319	272	0.008227	0.00241	0.756668	0.40625
BestDownload	319	272	0.008227	0.00241	0.756668	0.40625
AdaptiveSpeed	319	272	0.008227	0.00241	0.756668	0.40625
BestDownload	318	280	0.008243	0	0.765135	0.65625
BestSecurity	318	280	0.008243	0	0.765135	0.65625
AdaptiveSecurity	318	280	0.008243	0	0.765135	0.65625

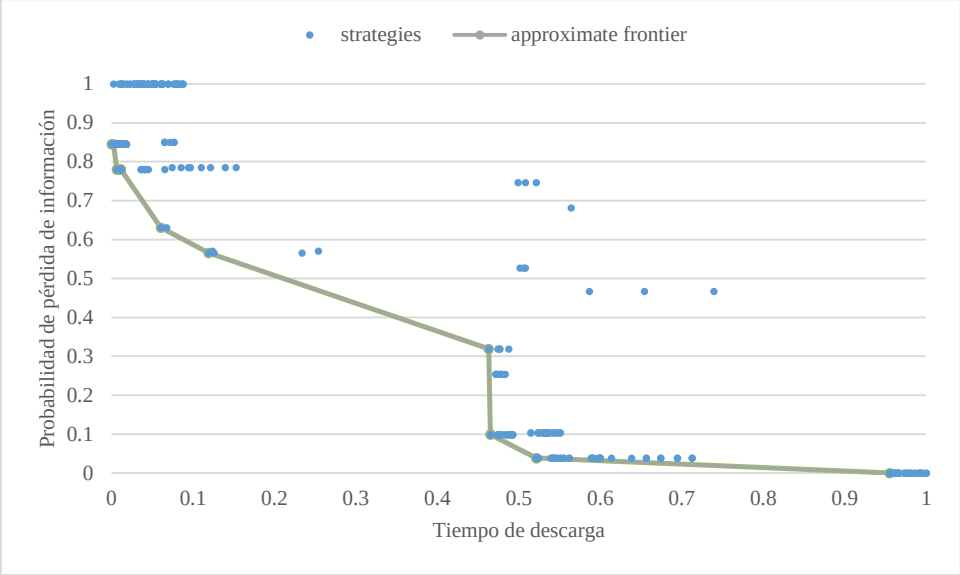

Figura 47. Aproximación de Pareto para la configuración $(2,2)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 45. Miembros del frente aproximado de Pareto de la configuración $(2,2)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	909	546	0.001519	0.441259	0	0.955056
Best Security	497	353	0.001605	0.209668	0.038699	0.521348
AdaptiveSecurity	158	353	0.001605	0.019112	0.038699	0.521348
Best Security	701	328	0.001738	0.32434	0.098895	0.465169
AdaptiveSecurity	138	328	0.001738	0.00787	0.098895	0.465169
Random	642	327	0.002225	0.291175	0.31904	0.462921
Random	208	174	0.00277	0.047218	0.565612	0.119101
Random	193	148	0.002914	0.038786	0.630812	0.060674
Random	143	126	0.003245	0.01068	0.780521	0.011236
BestUpload	143	126	0.003245	0.01068	0.780521	0.011236
AdaptiveSpeed	143	126	0.003245	0.01068	0.780521	0.011236
BestUpload	129	126	0.003245	0.002811	0.780521	0.011236
BestDownload	129	126	0.003245	0.002811	0.780521	0.011236
AdaptiveSpeed	129	126	0.003245	0.002811	0.780521	0.011236
BestDownload	140	124	0.003245	0.008994	0.780521	0.006742
BestUpload	124	124	0.003245	0	0.780521	0.006742
AdaptiveSpeed	124	124	0.003245	0	0.780521	0.006742
BestDownload	141	122	0.003389	0.009556	0.845297	0.002247
BestDownload	147	121	0.003389	0.012929	0.845297	0

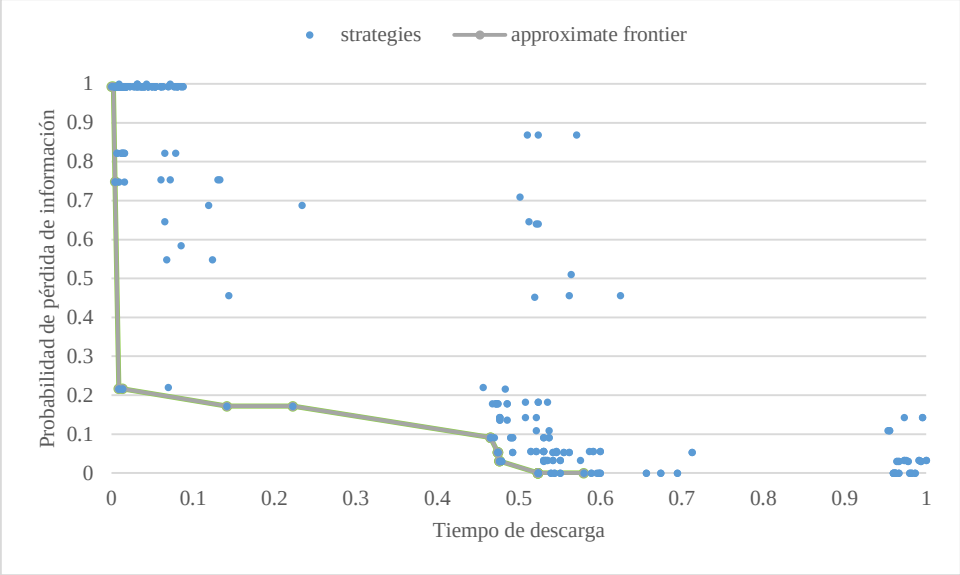

Figura 48. Aproximación de Pareto para la configuración $(2,3)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 46. Miembros del frente aproximado de Pareto de la configuración $(2,3)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	884	379	2.36E-06	0.352851	0	0.579775
BestSecurity	1043	354	2.36E-06	0.434515	6.43E-17	0.523596
BestSecurity	1261	354	2.36E-06	0.546482	6.43E-17	0.523596
AdaptiveSecurity	217	354	2.36E-06	0.010272	6.43E-17	0.523596
AdaptiveSecurity	216	354	2.36E-06	0.009759	6.43E-17	0.523596
Random	1996	333	2.56E-06	0.923986	0.030703	0.476404
BestSecurity	715	333	2.56E-06	0.26605	0.030703	0.476404
AdaptiveSecurity	237	333	2.56E-06	0.020544	0.030703	0.476404
BestSecurity	593	332	2.71E-06	0.20339	0.053283	0.474157
Random	736	332	2.71E-06	0.276836	0.053283	0.474157
AdaptiveSecurity	202	332	2.71E-06	0.002568	0.053283	0.474157
BestSecurity	833	328	2.96E-06	0.326656	0.091414	0.465169
AdaptiveSecurity	230	328	2.96E-06	0.016949	0.091414	0.465169
Random	558	220	3.49E-06	0.185413	0.171893	0.222472
Random	1109	184	3.49E-06	0.468413	0.171893	0.141573
Random	754	127	3.79E-06	0.286081	0.216851	0.013483
Random	470	125	3.79E-06	0.140216	0.216851	0.008989
BestDownload	329	123	7.29E-06	0.067797	0.748663	0.004494
BestDownload	210	122	8.91E-06	0.006677	0.993546	0.002247
BestDownload	219	121	8.91E-06	0.011299	0.993546	0

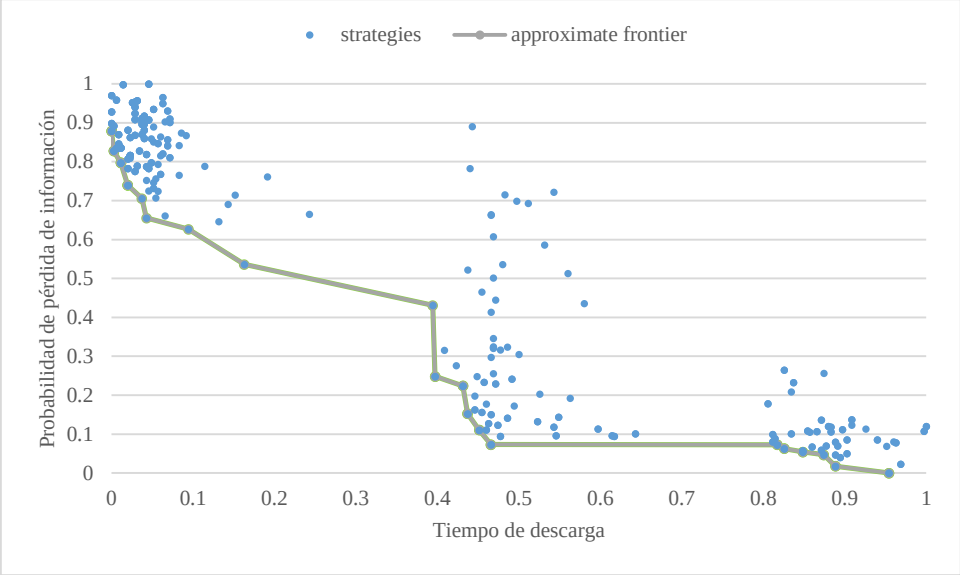

Figura 49. Aproximación de Pareto para la configuración $(3,3)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 47. Miembros del frente aproximado de Pareto de la configuración $(3,3)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	733	458	0.002359	0.464865	0	0.954286
AdaptiveSecurity	143	458	0.002359	0.009266	0	0.954286
BestSecurity	583	435	0.002407	0.349035	0.017738	0.888571
AdaptiveSecurity	161	435	0.002407	0.023166	0.017738	0.888571
BestSecurity	512	430	0.002486	0.294208	0.04678	0.874286
AdaptiveSecurity	145	430	0.002486	0.010811	0.04678	0.874286
BestSecurity	583	421	0.002507	0.349035	0.05446	0.848571
AdaptiveSecurity	145	421	0.002507	0.010811	0.05446	0.848571
BestSecurity	612	413	0.002531	0.371429	0.063027	0.825714
AdaptiveSecurity	166	413	0.002531	0.027027	0.063027	0.825714
BestSecurity	475	410	0.002557	0.265637	0.072783	0.817143
AdaptiveSecurity	157	410	0.002557	0.020077	0.072783	0.817143
BestSecurity	471	287	0.002559	0.262548	0.073238	0.465714
AdaptiveSecurity	159	287	0.002559	0.021622	0.073238	0.465714
BestSecurity	414	282	0.00266	0.218533	0.110431	0.451429
AdaptiveSecurity	152	282	0.00266	0.016216	0.110431	0.451429
BestSecurity	553	277	0.002775	0.325869	0.152552	0.437143
AdaptiveSecurity	153	277	0.002775	0.016988	0.152552	0.437143
BestSecurity	435	275	0.00297	0.234749	0.224223	0.431429
AdaptiveSecurity	166	275	0.00297	0.027027	0.224223	0.431429
BestSecurity	336	263	0.003035	0.158301	0.248208	0.397143
AdaptiveSecurity	154	263	0.003035	0.017761	0.248208	0.397143
Random	419	262	0.003533	0.222394	0.431059	0.394286
Random	220	181	0.00382	0.068726	0.536312	0.162857
Random	236	157	0.004065	0.081081	0.626486	0.094286
BestDownload	238	139	0.004145	0.082625	0.655606	0.042857
BestDownload	219	137	0.00428	0.067954	0.705407	0.037143
BestUpload	154	131	0.004374	0.017761	0.7398	0.02
BestDownload	154	131	0.004374	0.017761	0.7398	0.02
AdaptiveSpeed	154	131	0.004374	0.017761	0.7398	0.02
BestUpload	144	128	0.004531	0.010039	0.797377	0.011429
BestDownload	144	128	0.004531	0.010039	0.797377	0.011429
AdaptiveSpeed	144	128	0.004531	0.010039	0.797377	0.011429
BestUpload	160	125	0.004615	0.022394	0.828217	0.002857
BestDownload	160	125	0.004615	0.022394	0.828217	0.002857
AdaptiveSpeed	160	125	0.004615	0.022394	0.828217	0.002857
BestUpload	132	124	0.004754	0.000772	0.879201	0
BestDownload	132	124	0.004754	0.000772	0.879201	0
AdaptiveSpeed	132	124	0.004754	0.000772	0.879201	0

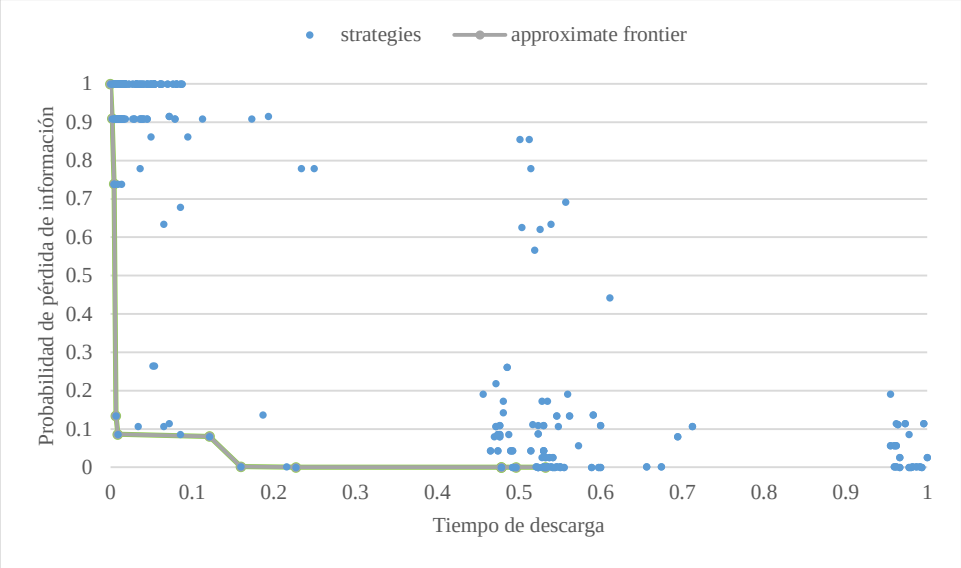

Figura 50. Aproximación de Pareto para la configuración $(2,4)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 48. Miembros del frente aproximado de Pareto de la configuración $(2,4)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	971	358	3.89E-09	0.321006	0	0.532584
Random	929	342	3.89E-09	0.300296	2.88E-09	0.496629
BestSecurity	826	334	3.89E-09	0.249507	2.22E-08	0.478652
AdaptiveSecurity	359	334	3.89E-09	0.019231	2.22E-08	0.478652
Random	973	222	3.89E-09	0.321992	4.44E-08	0.226966
Random	2126	192	3.92E-09	0.890533	0.001893	0.159551
Random	1457	175	5.04E-09	0.560651	0.080786	0.121348
Random	856	125	5.11E-09	0.2643	0.085831	0.008989
Random	1619	124	5.8E-09	0.640533	0.134508	0.006742
BestDownload	472	123	1.44E-08	0.074951	0.739263	0.004494
BestDownload	385	122	1.68E-08	0.032051	0.909472	0.002247
BestDownload	374	121	1.81E-08	0.026627	1	0

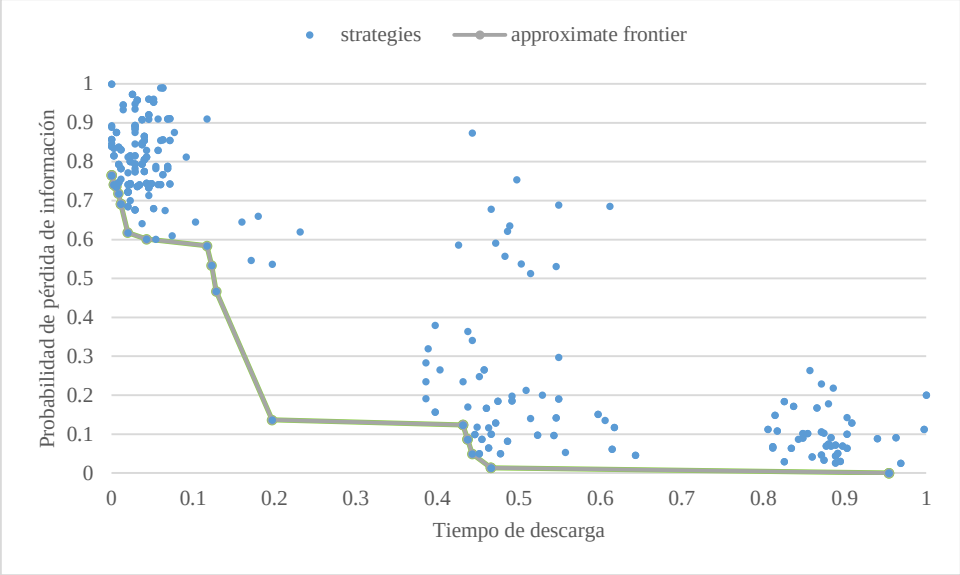

Figura 51. Aproximación de Pareto para la configuración $(3,4)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 49. Miembros del frente aproximado de Pareto de la configuración $(3,4)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	815	458	4.58E-06	0.446588	0	0.954286
AdaptiveSecurity	223	458	4.58E-06	0.007418	0	0.954286
BestSecurity	655	287	4.73E-06	0.327893	0.01355	0.465714
AdaptiveSecurity	253	287	4.73E-06	0.029674	0.01355	0.465714
Random	567	279	5.12E-06	0.262611	0.04952	0.442857
BestSecurity	689	277	5.54E-06	0.353116	0.086907	0.437143
AdaptiveSecurity	240	277	5.54E-06	0.02003	0.086907	0.437143
BestSecurity	678	275	5.95E-06	0.344955	0.123854	0.431429
AdaptiveSecurity	259	275	5.95E-06	0.034125	0.123854	0.431429
Random	442	193	6.09E-06	0.169881	0.136557	0.197143
Random	423	169	9.74E-06	0.155786	0.466894	0.128571
Random	428	167	1.05E-05	0.159496	0.533908	0.122857
Random	311	165	1.1E-05	0.0727	0.583795	0.117143
BestDownload	323	139	1.12E-05	0.081602	0.601032	0.042857
Random	371	131	1.14E-05	0.117211	0.618182	0.02
BestUpload	223	128	1.22E-05	0.007418	0.691723	0.011429
AdaptiveSpeed	223	128	1.22E-05	0.007418	0.691723	0.011429
BestDownload	288	127	1.25E-05	0.055638	0.718401	0.008571
BestUpload	227	126	1.27E-05	0.010386	0.736321	0.005714
AdaptiveSpeed	227	126	1.27E-05	0.010386	0.736321	0.005714
BestUpload	239	125	1.28E-05	0.019288	0.741567	0.002857
AdaptiveSpeed	239	125	1.28E-05	0.019288	0.741567	0.002857
BestUpload	214	124	1.3E-05	0.000742	0.76502	0
AdaptiveSpeed	214	124	1.3E-05	0.000742	0.76502	0

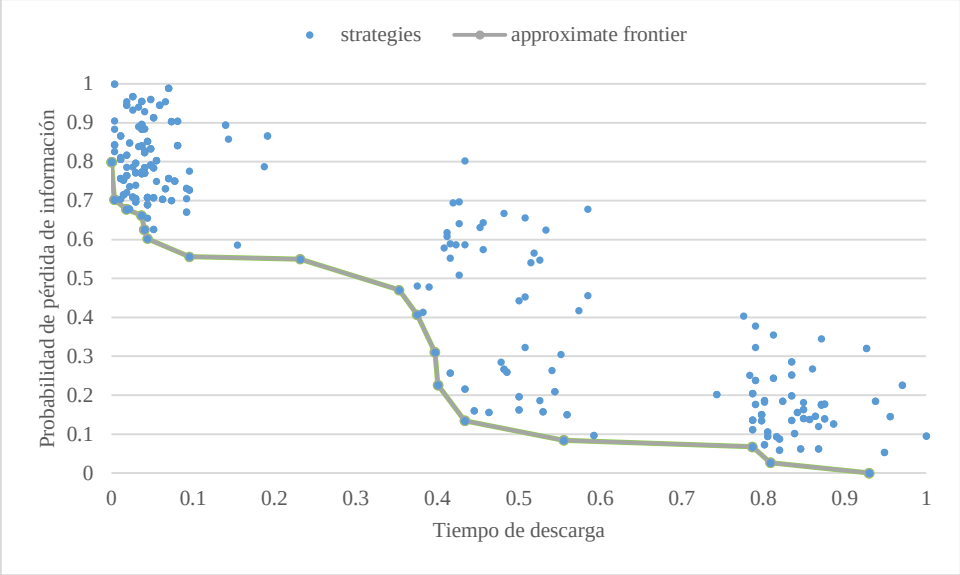

Figura 52. Aproximación de Pareto para la configuración $(4,4)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 50. Miembros del frente aproximado de Pareto de la configuración $(4,4)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	613	389	0.003258	0.446746	0	0.930147
AdaptiveSecurity	167	389	0.003258	0.006903	0	0.930147
BestSecurity	492	356	0.003341	0.327416	0.026701	0.808824
AdaptiveSecurity	190	356	0.003341	0.029586	0.026701	0.808824
BestSecurity	720	350	0.003467	0.552268	0.067315	0.786765
AdaptiveSecurity	183	350	0.003467	0.022682	0.067315	0.786765
BestSecurity	354	287	0.00352	0.191321	0.084463	0.555147
AdaptiveSecurity	170	287	0.00352	0.009862	0.084463	0.555147
BestSecurity	518	254	0.003676	0.353057	0.13461	0.433824
AdaptiveSecurity	181	254	0.003676	0.02071	0.13461	0.433824
BestSecurity	467	245	0.00396	0.302761	0.225935	0.400735
AdaptiveSecurity	171	245	0.00396	0.010848	0.225935	0.400735
BestSecurity	388	244	0.004224	0.224852	0.31061	0.397059
AdaptiveSecurity	189	244	0.004224	0.0286	0.31061	0.397059
Random	363	238	0.004524	0.200197	0.407218	0.375
Random	359	232	0.004721	0.196252	0.470304	0.352941
Random	233	199	0.004968	0.071992	0.549691	0.231618
Random	228	162	0.004987	0.067061	0.555847	0.095588
BestDownload	243	148	0.005131	0.081854	0.602283	0.044118
BestUpload	183	147	0.005205	0.022682	0.62593	0.040441
BestDownload	183	147	0.005205	0.022682	0.62593	0.040441
AdaptiveSpeed	183	147	0.005205	0.022682	0.62593	0.040441
BestUpload	176	146	0.005317	0.015779	0.661908	0.036765
AdaptiveSpeed	176	146	0.005317	0.015779	0.661908	0.036765
BestUpload	186	141	0.005366	0.025641	0.677848	0.018382
BestDownload	186	141	0.005366	0.025641	0.677848	0.018382
AdaptiveSpeed	186	141	0.005366	0.025641	0.677848	0.018382
BestUpload	170	137	0.005444	0.009862	0.702718	0.003676
BestDownload	170	137	0.005444	0.009862	0.702718	0.003676
AdaptiveSpeed	170	137	0.005444	0.009862	0.702718	0.003676
BestUpload	185	136	0.005744	0.024655	0.799368	0
BestDownload	185	136	0.005744	0.024655	0.799368	0
AdaptiveSpeed	185	136	0.005744	0.024655	0.799368	0

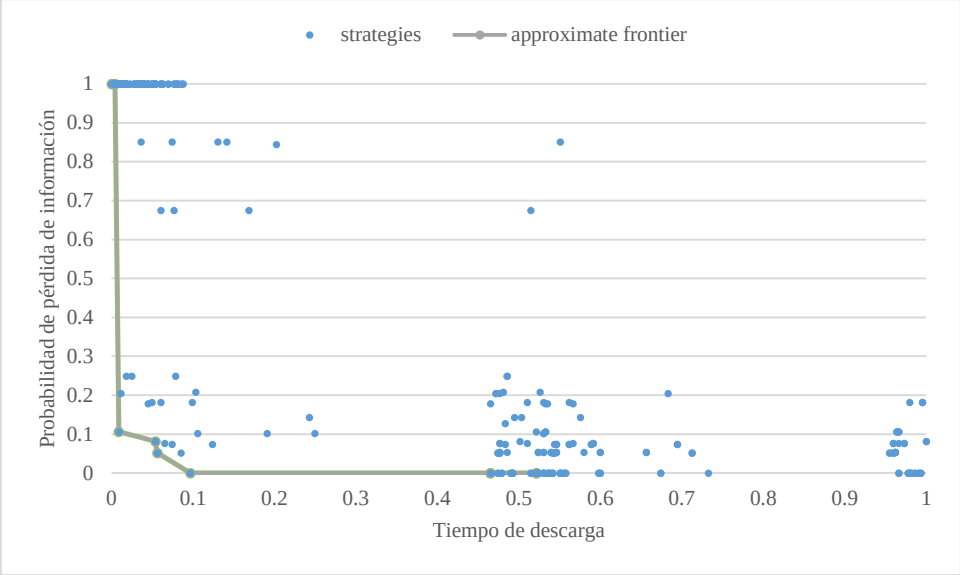

Figura 53. Aproximación de Pareto para la configuración $(2,5)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 51. Miembros del frente aproximado de Pareto de la configuración $(2,5)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	1083	353	6.38E-12	0.321974	0	0.521348
AdaptiveSecurity	555	353	6.38E-12	0.053408	0	0.521348
BestSecurity	1337	328	6.38E-12	0.45117	3.24E-17	0.465169
AdaptiveSecurity	566	328	6.38E-12	0.059003	3.24E-17	0.465169
Random	1272	164	6.38E-12	0.418108	2.11E-08	0.096629
Random	1205	146	7.67E-12	0.384028	0.051792	0.05618
Random	1464	145	8.41E-12	0.515768	0.081418	0.053933
Random	883	125	9.02E-12	0.220244	0.105958	0.008989
BestDownload	536	123	3.13E-11	0.043744	1	0.004494
BestDownload	479	122	3.13E-11	0.014751	1	0.002247
BestDownload	507	121	3.13E-11	0.028993	1	0

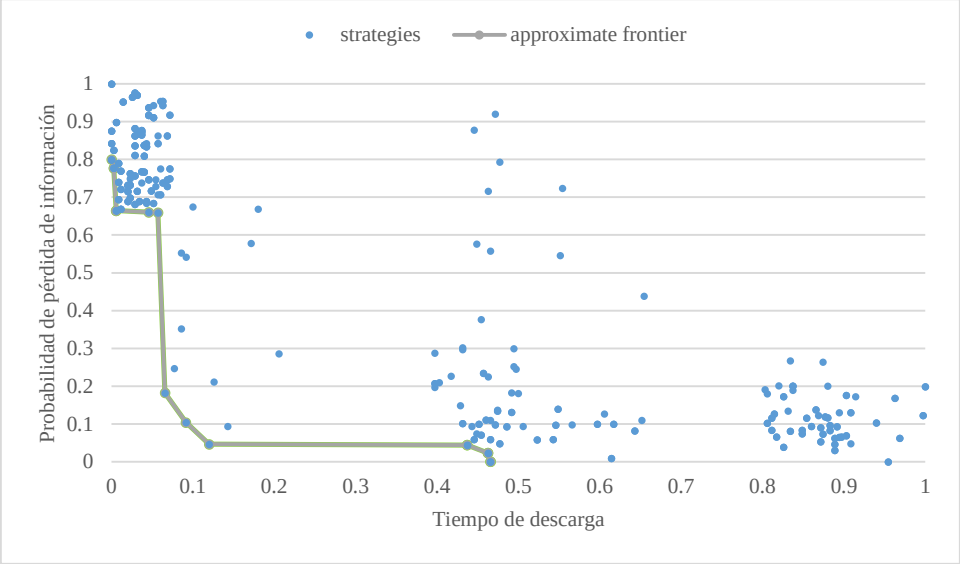

Figura 54. Aproximación de Pareto para la configuración $(3,5)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 52. Miembros del frente aproximado de Pareto de la configuración $(3,5)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	751	287	8.37E-09	0.345566	0	0.465714
AdaptiveSecurity	349	287	8.37E-09	0.038226	0	0.465714
BestSecurity	846	286	9E-09	0.418196	0.022908	0.462857
AdaptiveSecurity	325	286	9E-09	0.019878	0.022908	0.462857
BestSecurity	889	277	9.59E-09	0.45107	0.044249	0.437143
AdaptiveSecurity	376	277	9.59E-09	0.058869	0.044249	0.437143
Random	775	166	9.64E-09	0.363914	0.046028	0.12
Random	700	156	1.12E-08	0.306575	0.103979	0.091429
Random	732	147	1.34E-08	0.33104	0.182261	0.065714
Random	429	144	2.64E-08	0.099388	0.65918	0.057143
Random	501	140	2.65E-08	0.154434	0.660219	0.045714
BestUpload	341	126	2.66E-08	0.03211	0.664375	0.005714
AdaptiveSpeed	341	126	2.66E-08	0.03211	0.664375	0.005714
BestUpload	359	125	2.97E-08	0.045872	0.777316	0.002857
AdaptiveSpeed	359	125	2.97E-08	0.045872	0.777316	0.002857
BestDownload	312	124	3.03E-08	0.009939	0.799591	0

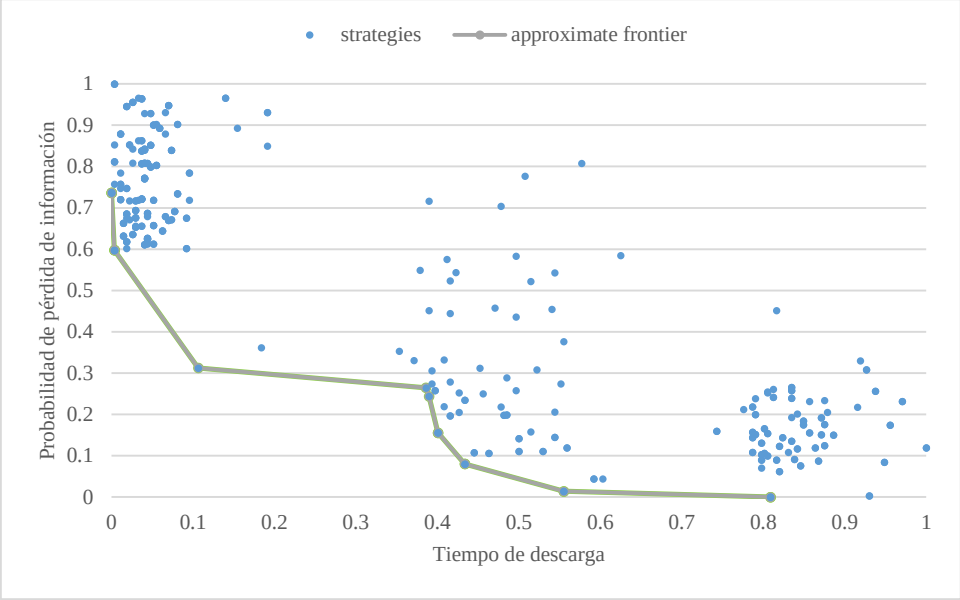

Figura 55. Aproximación de Pareto para la configuración $(4,5)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 53. Miembros del frente aproximado de Pareto de la configuración $(4,5)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	564	356	7.8E-06	0.344863	0	0.808824
AdaptiveSecurity	262	356	7.8E-06	0.03764	0	0.808824
BestSecurity	532	287	8.01E-06	0.312309	0.014016	0.555147
AdaptiveSecurity	257	287	8.01E-06	0.032553	0.014016	0.555147
BestSecurity	668	254	8.99E-06	0.450661	0.080184	0.433824
AdaptiveSecurity	283	254	8.99E-06	0.059003	0.080184	0.433824
BestSecurity	621	245	1.01E-05	0.402848	0.155506	0.400735
AdaptiveSecurity	267	245	1.01E-05	0.042726	0.155506	0.400735
Random	446	242	1.14E-05	0.224822	0.244034	0.389706
Random	644	241	1.17E-05	0.426246	0.264699	0.386029
Random	444	165	1.24E-05	0.222787	0.312486	0.106618
BestUpload	257	137	1.67E-05	0.032553	0.597631	0.003676
BestDownload	257	137	1.67E-05	0.032553	0.597631	0.003676
AdaptiveSpeed	257	137	1.67E-05	0.032553	0.597631	0.003676
BestUpload	270	136	1.87E-05	0.045778	0.736456	0
BestDownload	270	136	1.87E-05	0.045778	0.736456	0
AdaptiveSpeed	270	136	1.87E-05	0.045778	0.736456	0

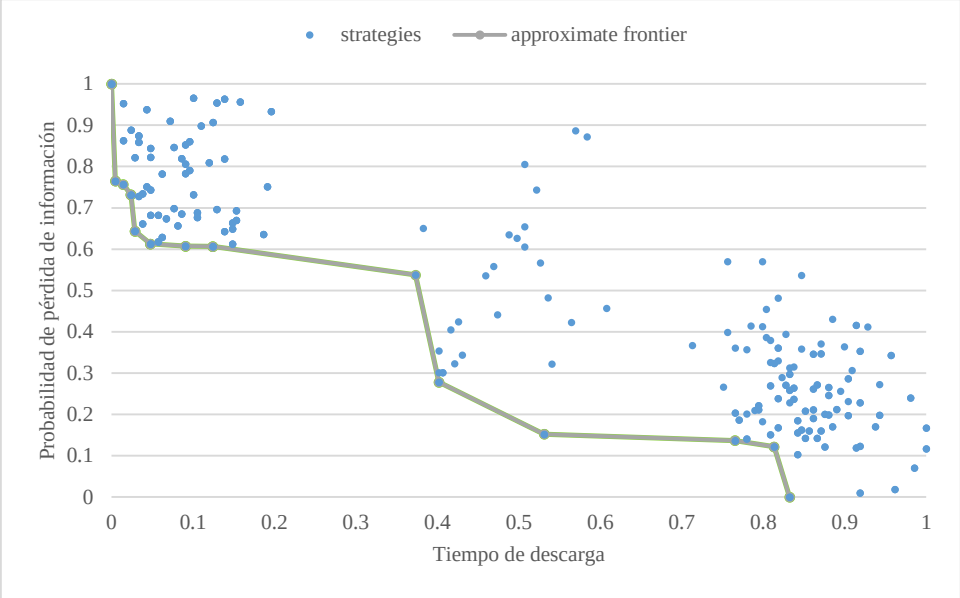

Figura 56. Aproximación de Pareto para la configuración (5,5). Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 54. Miembros del frente aproximado de Pareto de la configuración $(5,5)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	451	319	0.004232	0.344784	0	0.832536
AdaptiveSecurity	210	319	0.004232	0.038168	0	0.832536
BestSecurity	535	315	0.004622	0.451654	0.121619	0.813397
AdaptiveSecurity	226	315	0.004622	0.058524	0.121619	0.813397
BestSecurity	604	305	0.004671	0.53944	0.136637	0.76555
AdaptiveSecurity	210	305	0.004671	0.038168	0.136637	0.76555
BestSecurity	457	256	0.004721	0.352417	0.15226	0.5311
AdaptiveSecurity	240	256	0.004721	0.076336	0.15226	0.5311
BestSecurity	316	229	0.005124	0.173028	0.277949	0.401914
AdaptiveSecurity	232	229	0.005124	0.066158	0.277949	0.401914
Random	340	223	0.005956	0.203562	0.537453	0.373206
BestUpload	205	171	0.006178	0.031807	0.606413	0.124402
BestDownload	205	171	0.006178	0.031807	0.606413	0.124402
AdaptiveSpeed	205	171	0.006178	0.031807	0.606413	0.124402
BestUpload	183	164	0.006181	0.003817	0.607458	0.090909
BestDownload	183	164	0.006181	0.003817	0.607458	0.090909
AdaptiveSpeed	183	164	0.006181	0.003817	0.607458	0.090909
BestUpload	220	155	0.006198	0.050891	0.612684	0.047847
BestDownload	220	155	0.006198	0.050891	0.612684	0.047847
AdaptiveSpeed	220	155	0.006198	0.050891	0.612684	0.047847
BestUpload	226	151	0.006297	0.058524	0.643695	0.028708
BestDownload	226	151	0.006297	0.058524	0.643695	0.028708
AdaptiveSpeed	226	151	0.006297	0.058524	0.643695	0.028708
BestUpload	182	150	0.006579	0.002545	0.731585	0.023923
BestDownload	182	150	0.006579	0.002545	0.731585	0.023923
AdaptiveSpeed	182	150	0.006579	0.002545	0.731585	0.023923
BestUpload	219	148	0.00666	0.049618	0.756827	0.014354
AdaptiveSpeed	219	148	0.00666	0.049618	0.756827	0.014354
BestUpload	187	146	0.006686	0.008906	0.764824	0.004785
BestDownload	187	146	0.006686	0.008906	0.764824	0.004785
AdaptiveSpeed	187	146	0.006686	0.008906	0.764824	0.004785
BestUpload	196	145	0.00744	0.020356	1	0
BestDownload	196	145	0.00744	0.020356	1	0
AdaptiveSpeed	196	145	0.00744	0.020356	1	0

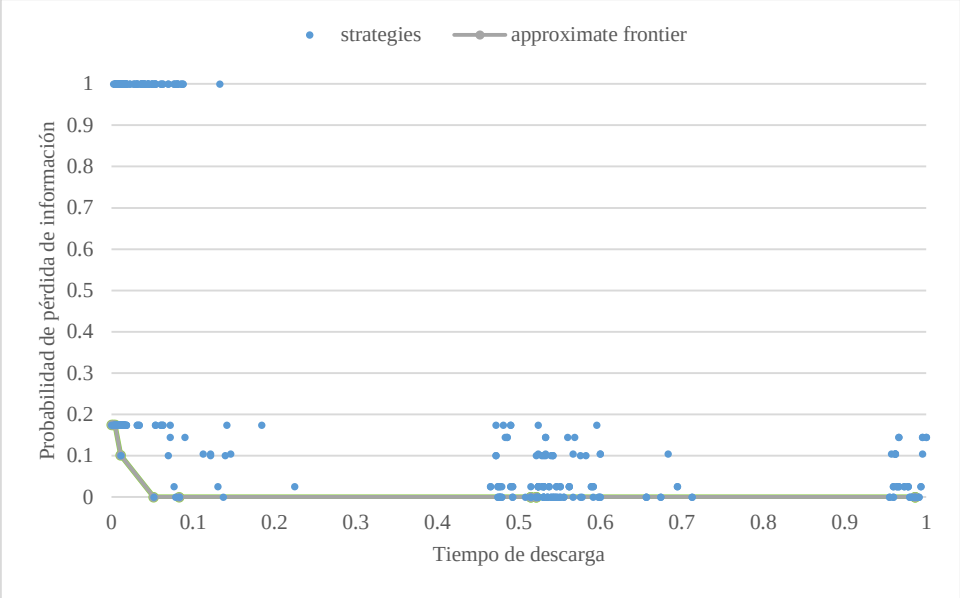

Figura 57. Aproximación de Pareto para la configuración $(2,6)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 55. Miembros del frente aproximado de Pareto de la configuración $(2,6)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	1376	560	1.24E-14	0.360432	0	0.986517
AdaptiveSecurity	818	560	1.24E-14	0.043206	0	0.986517
BestSecurity	1175	353	1.24E-14	0.246163	1.68E-08	0.521348
AdaptiveSecurity	870	353	1.24E-14	0.072769	1.68E-08	0.521348
BestSecurity	1474	350	1.24E-14	0.416146	2.5E-08	0.514607
AdaptiveSecurity	898	350	1.24E-14	0.088687	2.5E-08	0.514607
Random	1115	158	1.24E-14	0.212052	4.56E-08	0.083146
Random	1820	144	1.24E-14	0.612848	4.56E-08	0.051685
Random	1182	126	1.61E-14	0.250142	0.101257	0.011236
BestDownload	929	123	1.87E-14	0.10631	0.174683	0.004494
BestDownload	821	122	1.87E-14	0.044912	0.174683	0.002247
BestDownload	839	121	1.87E-14	0.055145	0.174683	0

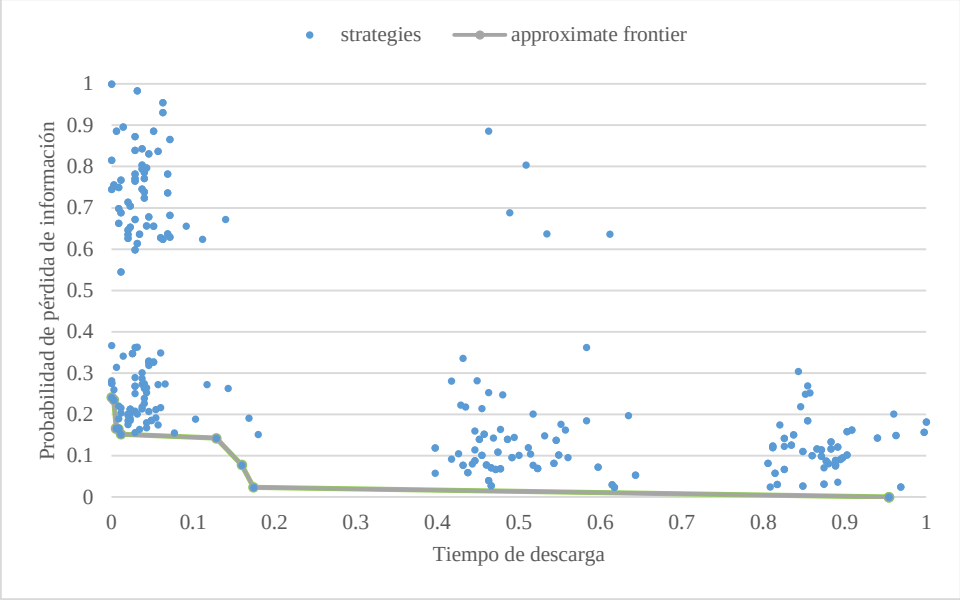

Figura 58. Aproximación de Pareto para la configuración $(3,6)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 56. Miembros del frente aproximado de Pareto de la configuración $(3,6)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	915	458	1.69E-11	0.360684	0	0.954286
AdaptiveSecurity	544	458	1.69E-11	0.04359	0	0.954286
Random	752	185	1.82E-11	0.221368	0.023477	0.174286
Random	1080	180	2.1E-11	0.501709	0.077707	0.16
Random	1531	169	2.44E-11	0.887179	0.14232	0.128571
BestDownload	720	128	2.49E-11	0.194017	0.15213	0.011429
BestDownload	805	127	2.56E-11	0.266667	0.164544	0.008571
BestUpload	561	126	2.57E-11	0.05812	0.166519	0.005714
BestDownload	561	126	2.57E-11	0.05812	0.166519	0.005714
AdaptiveSpeed	561	126	2.57E-11	0.05812	0.166519	0.005714
BestDownload	792	125	2.93E-11	0.255556	0.235201	0.002857
BestDownload	669	124	2.96E-11	0.150427	0.241302	0

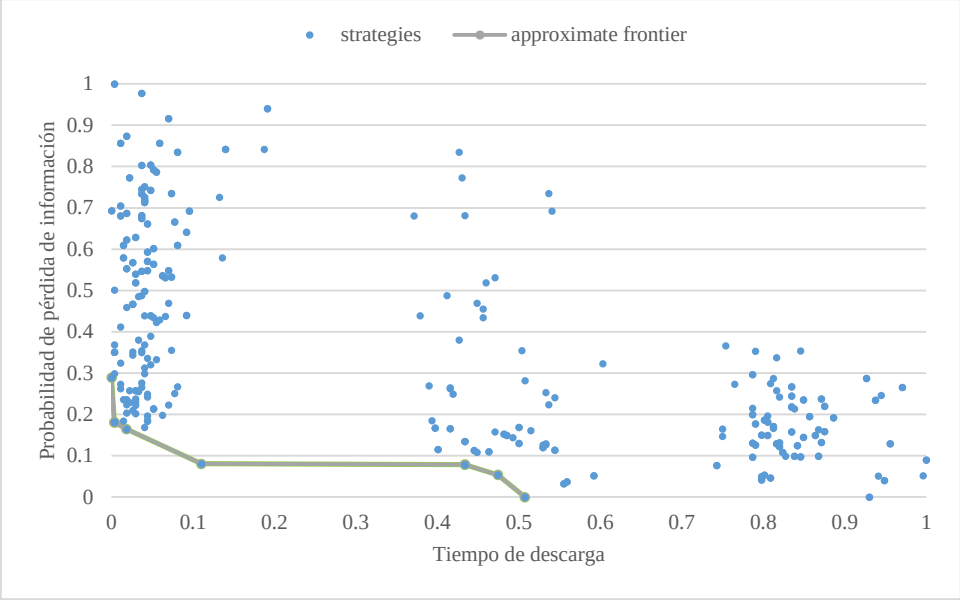

Figura 59. Aproximación de Pareto para la configuración $(4,6)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 57. Miembros del frente aproximado de Pareto de la configuración $(4,6)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	688	274	1.84E-08	0.360637	0	0.507353
Random	534	265	2.07E-08	0.185438	0.053745	0.474265
BestSecurity	702	254	2.17E-08	0.376564	0.078461	0.433824
AdaptiveSecurity	433	254	2.17E-08	0.070535	0.078461	0.433824
Random	662	166	2.18E-08	0.331058	0.080708	0.110294
BestDownload	541	141	2.53E-08	0.193402	0.164625	0.018382
BestUpload	422	137	2.6E-08	0.05802	0.181186	0.003676
BestDownload	422	137	2.6E-08	0.05802	0.181186	0.003676
AdaptiveSpeed	422	137	2.6E-08	0.05802	0.181186	0.003676
BestDownload	595	136	3.05E-08	0.254835	0.289402	0

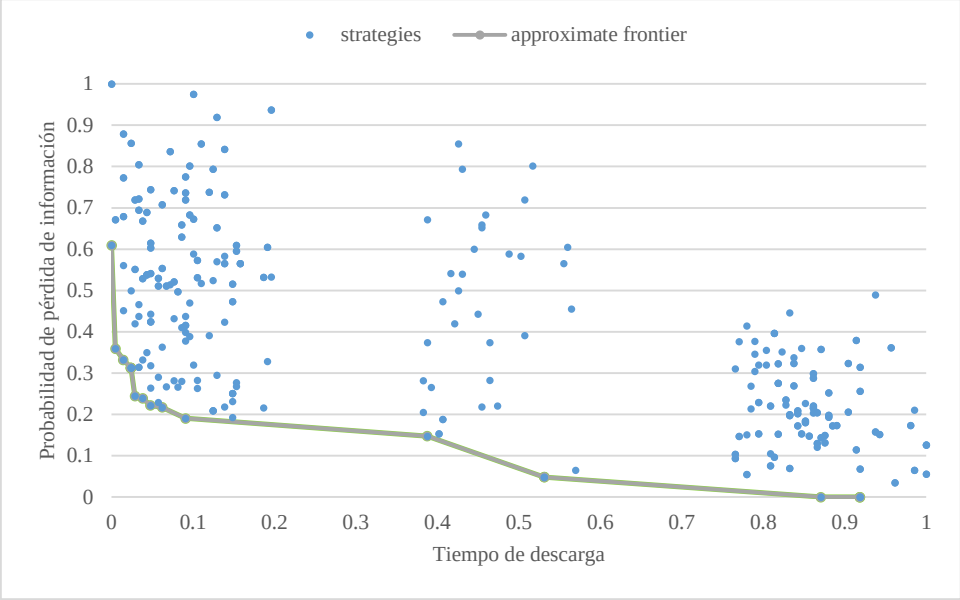

Figura 60. Aproximación de Pareto para la configuración $(5,6)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 58. Miembros del frente aproximado de Pareto de la configuración $(5,6)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	550	337	1.33E-05	0.357345	0	0.91866
AdaptiveSecurity	327	337	1.33E-05	0.042373	0	0.91866
Random	550	327	1.33E-05	0.357345	1.95E-16	0.870813
BestSecurity	607	256	1.41E-05	0.437853	0.048464	0.5311
AdaptiveSecurity	390	256	1.41E-05	0.131356	0.048464	0.5311
Random	615	226	1.58E-05	0.449153	0.147727	0.38756
BestDownload	433	164	1.66E-05	0.19209	0.19049	0.090909
BestDownload	396	158	1.71E-05	0.139831	0.21726	0.062201
BestDownload	430	155	1.71E-05	0.187853	0.221732	0.047847
BestDownload	335	153	1.74E-05	0.053672	0.239144	0.038278
BestDownload	470	151	1.75E-05	0.24435	0.244809	0.028708
BestUpload	331	150	1.87E-05	0.048023	0.312823	0.023923
BestDownload	331	150	1.87E-05	0.048023	0.312823	0.023923
AdaptiveSpeed	331	150	1.87E-05	0.048023	0.312823	0.023923
BestDownload	352	148	1.91E-05	0.077684	0.332452	0.014354
BestDownload	402	146	1.95E-05	0.148305	0.359239	0.004785
BestDownload	399	145	2.39E-05	0.144068	0.609744	0

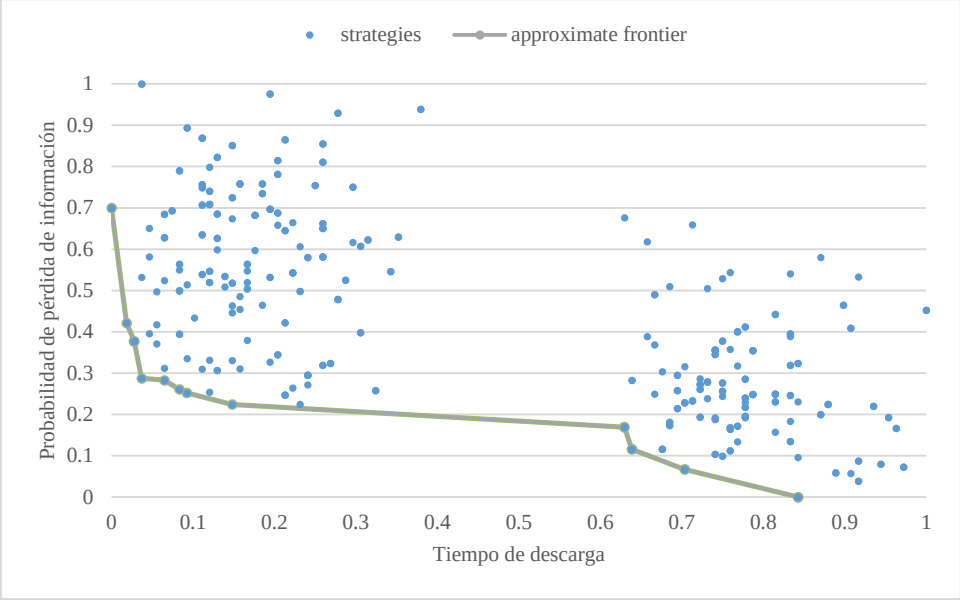

Figura 61. Aproximación de Pareto para la configuración $(6,6)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 59. Miembros del frente aproximado de Pareto de la configuración $(6,6)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	454	299	0.005514	0.360345	0	0.842593
AdaptiveSecurity	270	299	0.005514	0.043103	0	0.842593
BestSecurity	375	284	0.005715	0.224138	0.066988	0.703704
AdaptiveSecurity	273	284	0.005715	0.048276	0.066988	0.703704
BestSecurity	529	277	0.00586	0.489655	0.115796	0.638889
AdaptiveSecurity	295	277	0.00586	0.086207	0.115796	0.638889
BestSecurity	507	276	0.00602	0.451724	0.169225	0.62963
AdaptiveSecurity	288	276	0.00602	0.074138	0.169225	0.62963
BestDownload	357	224	0.006185	0.193103	0.224303	0.148148
BestDownload	355	218	0.006268	0.189655	0.252022	0.092593
BestDownload	307	217	0.006294	0.106897	0.260699	0.083333
BestDownload	277	215	0.006359	0.055172	0.282583	0.064815
BestDownload	388	212	0.006373	0.246552	0.287355	0.037037
BestUpload	273	211	0.006642	0.048276	0.377291	0.027778
BestDownload	273	211	0.006642	0.048276	0.377291	0.027778
AdaptiveSpeed	273	211	0.006642	0.048276	0.377291	0.027778
BestDownload	332	210	0.006774	0.15	0.421364	0.018519
BestDownload	329	208	0.007607	0.144828	0.700029	0

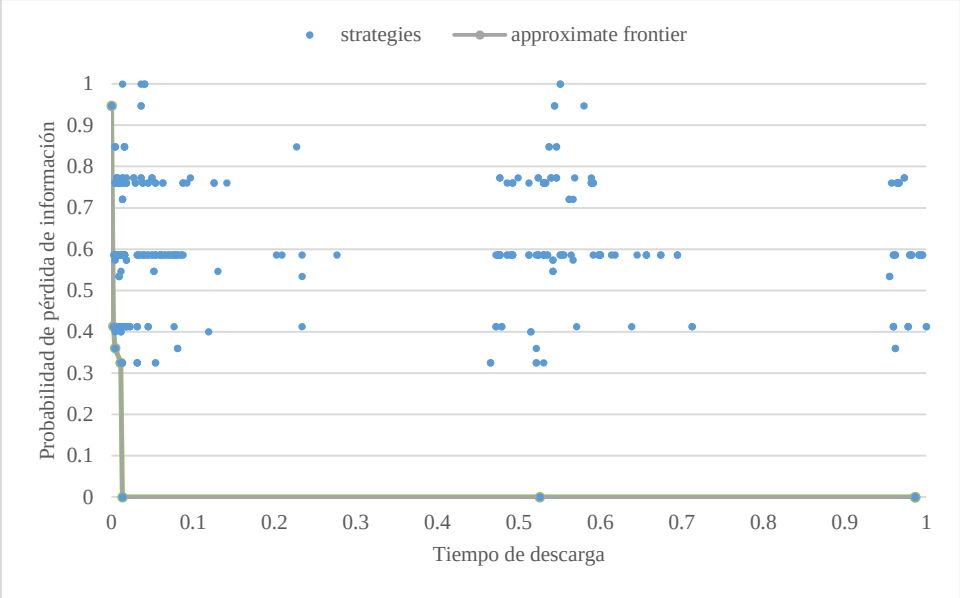

Figura 62. Aproximación de Pareto para la configuración $(2,7)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 60. Miembros del frente aproximado de Pareto de la configuración (2,7) para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	1443	560	2.59E-17	0.223061	0	0.986517
AdaptiveSecurity	1443	560	2.59E-17	0.223061	0	0.986517
Random	1443	355	2.59E-17	0.223061	3.7E-10	0.525843
BestDownload	1443	127	2.59E-17	0.223061	7.39E-10	0.013483
BestDownload	1242	126	2.59E-17	0.085106	0.325183	0.011236
BestDownload	1205	123	2.59E-17	0.059712	0.360536	0.004494
BestDownload	1118	122	2.59E-17	0	0.413064	0.002247
BestDownload	1145	121	2.59E-17	0.018531	0.947472	0

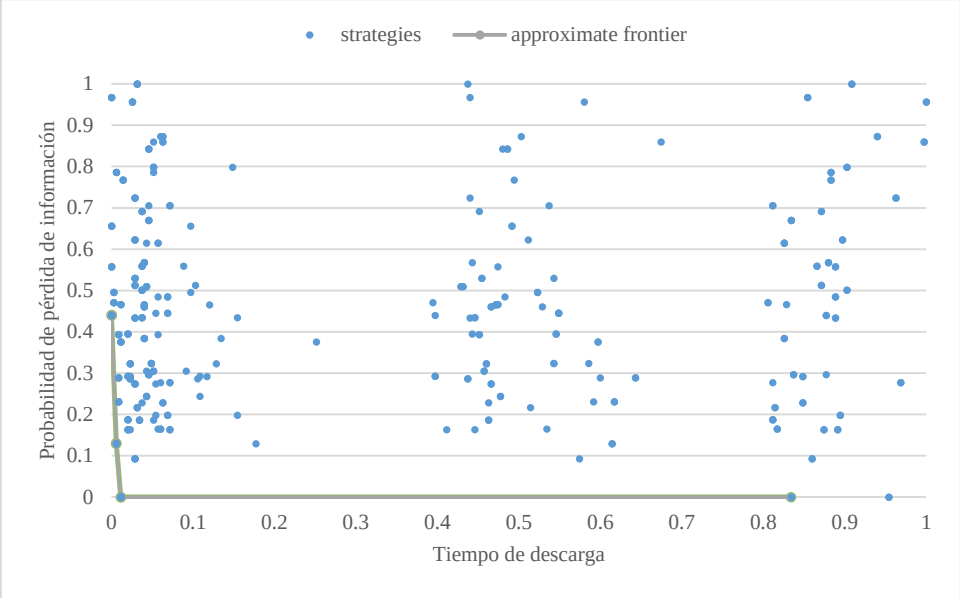

Figura 63. Aproximación de Pareto para la configuración $(3,7)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 61. Miembros del frente aproximado de Pareto de la configuración $(3,7)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	959	416	3.5E-14	0.22291	0	0.834286
BestUpload	959	128	3.5E-14	0.22291	2.66E-16	0.011429
AdaptiveSpeed	959	128	3.5E-14	0.22291	2.66E-16	0.011429
BestUpload	798	126	3.81E-14	0.05676	0.129803	0.005714
BestDownload	798	126	3.81E-14	0.05676	0.129803	0.005714
AdaptiveSpeed	798	126	3.81E-14	0.05676	0.129803	0.005714
BestUpload	924	124	4.55E-14	0.186791	0.44038	0
BestDownload	924	124	4.55E-14	0.186791	0.44038	0
AdaptiveSpeed	924	124	4.55E-14	0.186791	0.44038	0

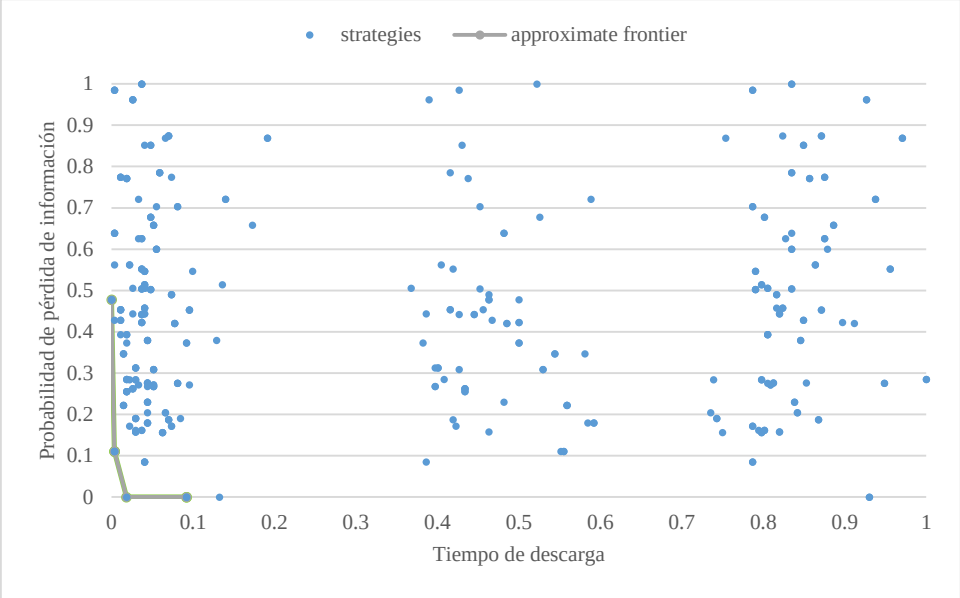

Figura 64. Aproximación de Pareto para la configuración $(4,7)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 62. Miembros del frente aproximado de Pareto de la configuración $(4,7)$ para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestUpload	721	161	4.17E-11	0.222527	0	0.091912
AdaptiveSpeed	721	161	4.17E-11	0.222527	0	0.091912
BestDownload	721	141	4.17E-11	0.222527	1.71E-16	0.018382
BestUpload	600	137	4.59E-11	0.056319	0.110898	0.003676
BestDownload	600	137	4.59E-11	0.056319	0.110898	0.003676
AdaptiveSpeed	600	137	4.59E-11	0.056319	0.110898	0.003676
BestDownload	774	136	5.97E-11	0.29533	0.478095	0

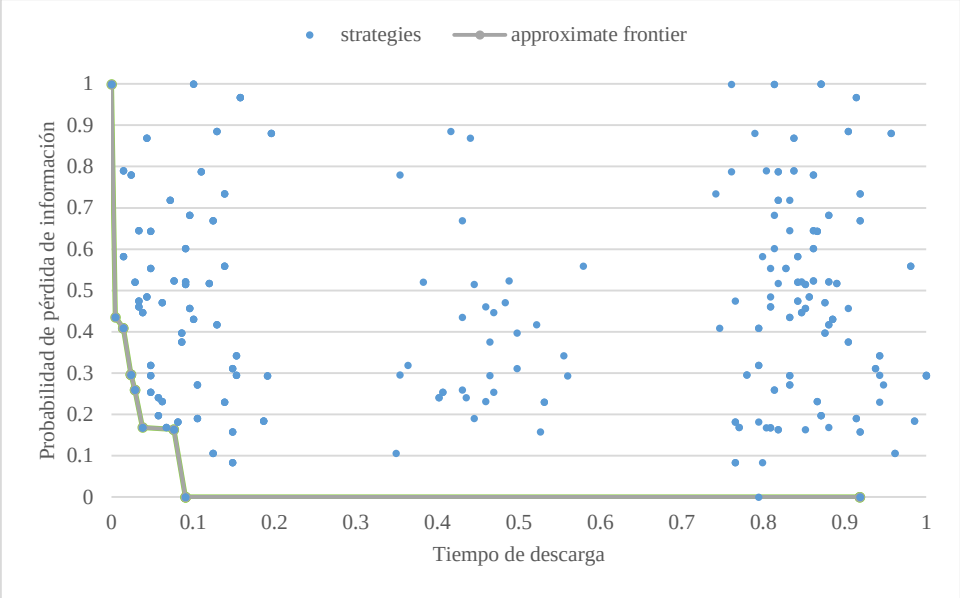

Figura 65. Aproximación de Pareto para la configuración $(5,7)$. Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 63. Miembros del frente aproximado de Pareto de la configuración (5,7) para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
BestSecurity	577	337	3.61E-08	0.222985	0	0.91866
AdaptiveSecurity	577	337	3.61E-08	0.222985	0	0.91866
BestUpload	577	164	3.61E-08	0.222985	2.27E-16	0.090909
BestDownload	577	164	3.61E-08	0.222985	2.27E-16	0.090909
AdaptiveSpeed	577	164	3.61E-08	0.222985	2.27E-16	0.090909
BestUpload	565	161	4.08E-08	0.202401	0.163316	0.076555
AdaptiveSpeed	565	161	4.08E-08	0.202401	0.163316	0.076555
BestDownload	458	153	4.1E-08	0.018868	0.168511	0.038278
BestUpload	590	151	4.36E-08	0.245283	0.259385	0.028708
AdaptiveSpeed	590	151	4.36E-08	0.245283	0.259385	0.028708
BestUpload	481	150	4.47E-08	0.058319	0.296267	0.023923
BestDownload	481	150	4.47E-08	0.058319	0.296267	0.023923
AdaptiveSpeed	481	150	4.47E-08	0.058319	0.296267	0.023923
BestUpload	480	148	4.8E-08	0.056604	0.408925	0.014354
AdaptiveSpeed	480	148	4.8E-08	0.056604	0.408925	0.014354
BestDownload	556	146	4.88E-08	0.186964	0.43536	0.004785
BestDownload	546	145	6.52E-08	0.169811	0.999425	0

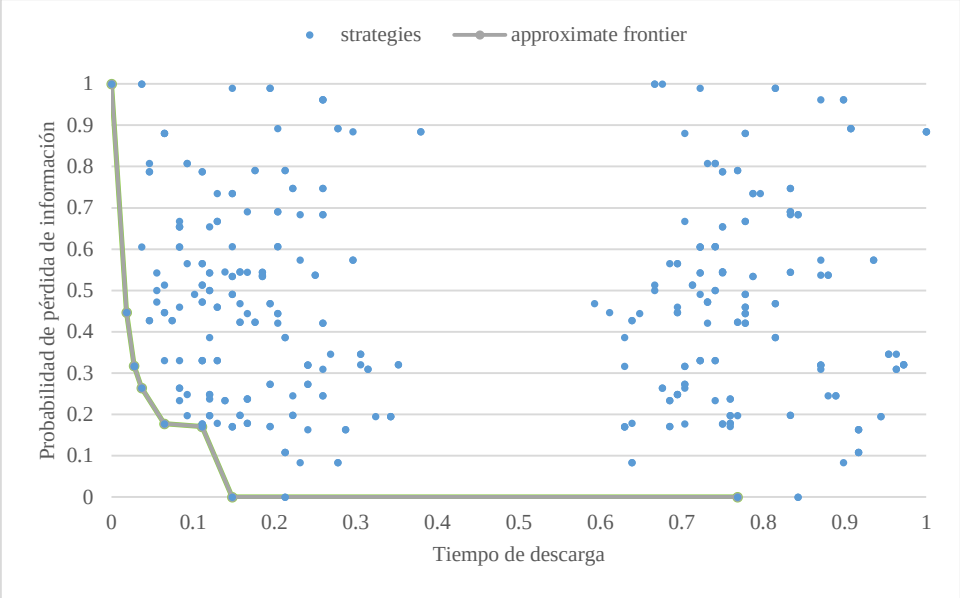

Figura 66. Aproximación de Pareto para la configuración (6,7). Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 64. Miembros del frente aproximado de Pareto de la configuración (6,7) para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	476	291	2.05E-05	0.222917	0	0.768519
BestDownload	476	224	2.05E-05	0.222917	2.92E-16	0.148148
BestDownload	559	220	2.25E-05	0.395833	0.170203	0.111111
BestDownload	377	215	2.25E-05	0.016667	0.177329	0.064815
BestDownload	487	212	2.35E-05	0.245833	0.264288	0.037037
BestDownload	397	211	2.42E-05	0.058333	0.316992	0.027778
BestDownload	458	210	2.57E-05	0.185417	0.44703	0.018519
BestDownload	450	208	3.21E-05	0.16875	1	0

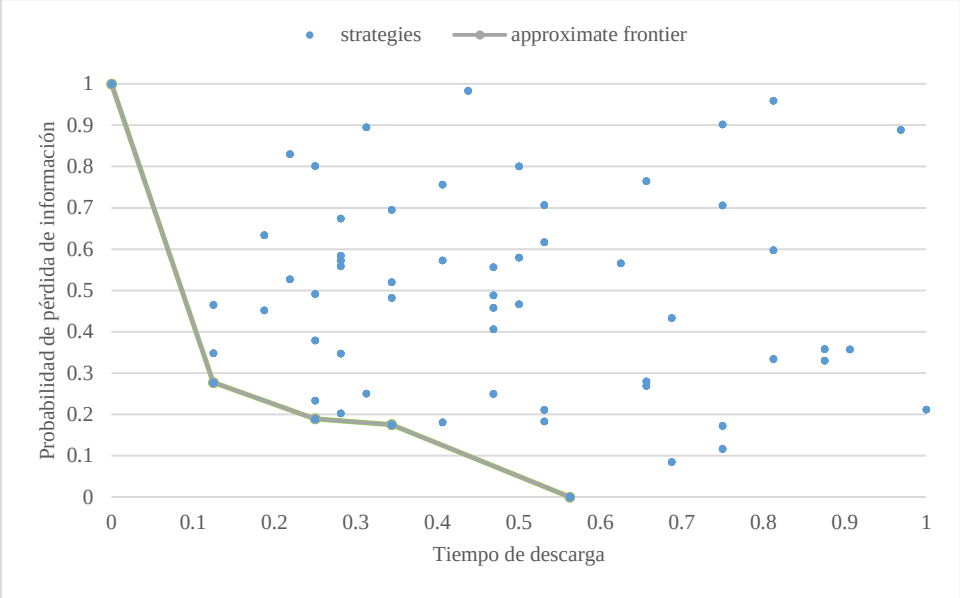

Figura 67. Aproximación de Pareto para la configuración (7,7). Probabilidad de pérdida de información vs tiempo de descarga.

Tabla 65. Miembros del frente aproximado de Pareto de la configuración (7,7) para probabilidad de pérdida de información vs tiempo de descarga.

Estrategia	Tiempo de carga	Tiempo de descarga	Probabilidad de pérdida de información	Tiempo de carga normalizado	Probabilidad de falla normalizada	Tiempo de descarga normalizado
Random	411	277	0.00682	0.224096	0	0.5625
BestDownload	411	277	0.00682	0.224096	0	0.5625
Random	483	270	0.007146	0.39759	0.175386	0.34375
BestDownload	483	270	0.007146	0.39759	0.175386	0.34375
BestSecurity	483	270	0.007146	0.39759	0.175386	0.34375
AdaptiveSecurity	483	270	0.007146	0.39759	0.175386	0.34375
BestUpload	326	267	0.007172	0.019277	0.189358	0.25
BestDownload	326	267	0.007172	0.019277	0.189358	0.25
AdaptiveSpeed	326	267	0.007172	0.019277	0.189358	0.25
Random	420	263	0.007335	0.245783	0.277146	0.125
BestSecurity	420	263	0.007335	0.245783	0.277146	0.125
AdaptiveSecurity	420	263	0.007335	0.245783	0.277146	0.125
Random	389	259	0.00868	0.171084	1	0
BestSecurity	389	259	0.00868	0.171084	1	0
AdaptiveSecurity	389	259	0.00868	0.171084	1	0