

TESIS

Presentada el día 16 de noviembre de 2009 en la

Universidad de La República, UdelaR

para obtener el título de

MAGISTER EN INGENIERÍA
Mención INGENIERÍA MATEMÁTICA

para

Pablo Gabriel ROMERO RODRÍGUEZ

Instituto de Investigación : LPE - IMERL

Componentes universitarios :
UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA

Título de la tesis :

*Optimización de la Estrategia de Selección de Piezas de Video en
Redes P2P*

Comité de examinadores:

Dr. Franco	ROBLEDO	Director de Tesis
Dr. Pablo	RODRIGUEZ	Director de Tesis
Dr. Gerardo	RUBINO	Presidente
Dr. Federico	NAJSON	
Dr. Alvaro	MARTÍN	

Agradecimientos

Esta tesis fue realizada con placer por múltiples razones.

Los tutores que la dirigieron, Dr. Franco Robledo y Dr. Pablo Rodríguez-Bocca, supieron motivarme y deleitarme con cada nueva actividad vinculada con esta tesis, desde antes de sus inicios. También ofrecer su visión, guía y aportes permanentemente. A su vez, los licenciados en estadística Darío Padula y María Elisa Bertinat, han propuesto y promovido no sólo contribuciones sino también compañerismo. Gracias a los ingenieros Daniel de Vera, Matías Barrios y Andrés Barrios, las contribuciones de esta tesis toman valor agregado al aplicarse en una red real.

Agradezco siempre a mi abuela que me enseñó a sumar y restar de pequeño, mi madre que con gran cariño y dedicación me ha sabido criar, destacando siempre la importancia de estudiar. Esta tesis se ha escrito en Rennes, Francia. Agradezco profundamente el amor que me ha brindado mi novia a pesar de su distancia desde Uruguay, que es sin duda uno de los motores de la curiosidad e investigación.

Finalmente, agradezco a los doctores Gerardo Rubino, Alvaro Martín y Federico Najson, que me brindan el honor de juzgar este trabajo.

Índice general

Índice de Contenidos	1
I INTRODUCCIÓN	5
0.1. Publicaciones	12
II MARCO TEÓRICO	13
1. Distribución y Evaluación del Video	15
1.1. Introducción	15
1.2. Distintas concepciones del video digital	16
1.3. Redes de Entrega de Contenidos (VDN)	16
1.3.1. Televisión por Internet	16
1.3.2. P2PTV	16
1.3.3. IPTV	17
1.3.4. MobileTV	17
1.3.5. TV cable y TV Satelital	17
1.4. Calidad del video	18
1.4.1. Medidas objetivas y subjetivas	18
1.4.2. DCR (Degraded Category Rating)	19
1.4.3. Críticas a las Técnicas Históricas	19
1.5. Otro enfoque: PSQA	20
1.6. Parámetros de Impacto	21
1.6.1. Parámetros de ambiente	21
1.6.2. Parámetros de la fuente	21
1.6.3. Parámetros de distribución	22
1.7. Latencia y Continuidad en video streaming	22
1.7.1. Continuidad del Video	22
1.7.2. Latencia inicial del Video	23
1.8. Conclusiones	23

2. BitTorrent, GoalBit y Video en Vivo	25
2.1. Introducción	25
2.2. BitTorrent	25
2.2.1. Introducción	25
2.2.2. Descargar archivos con BitTorrent	26
2.2.3. Arquitectura de BitTorrent	26
2.2.4. Estrategia de Selección de Piezas	26
2.2.4.1. Política de Rarest First	26
2.2.5. Estrategia de Selección de Pares	27
2.2.5.1. Tit-for-tat	27
2.2.6. Optimistic Unchoking	27
2.2.7. Conclusiones Parciales	28
2.3. GoalBit	28
2.3.1. Introducción	28
2.3.2. Streaming de Video P2P	28
2.3.3. La Plataforma GoalBit	29
2.3.4. Protocolo de Transporte GoalBit	29
2.3.5. Información en el tracker	30
2.3.6. Archivos .goalbit	31
2.3.7. Comunicación Par-Tracker	31
2.3.8. Comunicación Entre Pares	31
2.3.8.1. Mensajes de control entre pares	33
2.3.8.2. Mensajes de Intercambio de Piezas	36
2.3.8.3. Comunicación exitosa entre pares	39
2.3.9. Estrategias aplicadas al Protocolo	40
2.3.9.1. Estrategia de Selección de pares	40
2.3.9.2. Estrategia de selección de piezas	40
2.4. Clásicas Estrategias de Selección de Piezas	41
2.5. Conclusiones	42
III MODELO DE COOPERACION EN REDES P2P	43
3. Modelo Matemático de Cooperación	45
3.1. Introducción	45
3.2. Modelo Matemático	46
3.3. Estrategias Clásicas y una Hibridación	49
3.4. Análisis de Desempeño	50
3.4.1. Performance de Rarest First	51
3.4.2. Performance de Greedy	53
3.4.3. Cotas de los Parámetros de Calidad	54
3.5. Análisis Primitivo de nuevas Estrategias	55
3.5.1. Greedy Ponderado	56
3.5.2. La estrategia Mixta	58

3.6. Conclusiones	59
4. Nuevas Estrategias de Selección de Piezas	61
4.1. Introducción	61
4.2. Familia de Permutaciones	62
4.2.1. Propiedades Básicas de la Familia	63
4.2.2. Experiencia en la Búsqueda de soluciones	65
4.3. Medidas de Optimalidad	72
4.3.1. Medida basada en la extensión de una consulta	73
4.3.2. Dos problemas de Optimización	75
4.3.3. Discusión sobre Optimalidad en ambos problemas	77
4.4. Formalización del Problema y Metaheurísticas	82
4.4.1. Introducción	82
4.4.2. Conceptos Básicos	83
4.4.2.1. Greedy	83
4.4.2.2. Búsqueda Local	84
4.4.2.3. Optimización por Colonia de Hormigas (ACO)	85
4.4.3. Problema de Optimización Combinatoria	86
4.5. Algoritmo de Búsqueda Local	87
4.5.1. Espacio Métrico con estructura de vecindad	87
4.5.2. Algoritmo de Búsqueda Local	89
4.6. Búsqueda de Permutación basada en ACO	90
4.6.1. Introducción	90
4.6.2. Biyección entre una permutación y un ciclo	90
4.6.3. El Algoritmo Principal	91
4.6.3.1. Función <i>InicializarAristas</i>	92
4.6.3.2. Función <i>InicializarFeromonas</i>	93
4.6.3.3. Implementación de ACO	95
4.6.3.4. Resultados y comparaciones	97
4.6.4. Análisis de la cantidad de operaciones	97
4.7. Conclusiones	101
IV DESEMPEÑO DE NUEVAS ESTRATEGIAS EN GOALBIT	103
5. Análisis de Performance en GoalBit	105
5.1. Introducción	105
5.2. Hipótesis asumidas VS Red real	106
5.2.1. Parámetros de desempeño en GoalBit	107
5.3. Emulación de GoalBit	108
5.3.1. Emulación VS Simulación de GoalBit	108
5.3.2. Emulación de GoalBit	109
5.3.3. Bloque 1: Configurar parámetros de entrada	109
5.3.4. Bloque 2: Levantar el tracker	110

5.3.5.	Bloque 3: Levantar los pares	111
5.3.6.	Bloque 4: Procesamiento estadístico	111
5.4.	Escenarios de emulación	112
5.4.1.	Propiedades del Ordenador	112
5.4.2.	Casos de prueba	112
5.5.	Resultados	112
5.5.1.	Instancia 1: 45 pares y 10 minutos	113
5.5.1.1.	Observaciones	114
5.5.2.	Instancia 2: 135 pares y 50 minutos	115
5.5.2.1.	Observaciones	115
5.5.3.	Instancia 3: 350 pares y 10 minutos	116
5.5.3.1.	Observaciones	117
5.5.4.	Resumen de Resultados	117
5.6.	Conclusiones	118
6.	CONCLUSIONES GENERALES	121
	Bibliografía	133
	Índice de Contenidos	133
	Lista de Figuras	134

Parte I

INTRODUCCIÓN

ABSTRACT

El tráfico debido a redes Peer-to-Peer (P2P) sobre la infraestructura de Internet es superior a la suma de las restantes aplicaciones. Si bien esto último es un gran desafío económico para los proveedores de servicios de Internet, las prestaciones de estas redes son tan importantes en la actualidad (interactividad y comunicación multimedia, descarga de contenidos de video, música, transacciones y trabajos cooperativos, entre tantos otros) que motivan a promover políticas de diseño tanto eficaces como eficientes.

Las redes P2P imponen un nuevo paradigma de comunicación en base a la cooperación entre pares, que se contrasta con los anteriores modelos cliente-servidor. Ahora los contenidos se fraccionan y distribuyen en la red. Los pares se benefician de los recursos de los restantes para lograr obtener el contenido entero. Hay dos elementos estratégicos claves en el diseño de redes de pares: la decisión de pares con los cuales cooperar, o “estrategia de selección de pares”, y las piezas que se deben descargar una vez establecida la comunicación con un par, o “estrategia de selección de piezas”.

Esta tesis se concentra en la estrategia de selección de piezas. En base a un modelo matemático simple, se proponen nuevas técnicas de estrategia de selección de piezas, que superan la calidad de anteriores técnicas. Se introduce un Problema de Optimización Combinatoria (COP) centrado en la calidad de experiencia (QoE), cuyo conjunto de variables de decisión es el de estrategias de selección de piezas. La experiencia en la resolución de este problema sugiere bondades en ciertas estrategias. Las mismas se introducen en una plataforma real de video-streaming denominada GoalBit, que es la primera red P2P cuyos protocolos y códigos son públicos, y posee amplia difusión actualmente en Internet.

Palabras Clave: P2P, COP, QoE, video-streaming.

INTRODUCCIÓN

Internet ha revolucionado las telecomunicaciones desde sus inicios. Hoy en día, los usuarios pueden disfrutar de las más variadas aplicaciones, que van desde un simple correo electrónico y transferencia de archivos, hasta descarga de videos, música, juegos en red, comunicación multimedia a distancia que puede incluir voz, video y texto alfanumérico, entre tantas otras.

Los servicios de banda ancha han penetrado en el mercado mundial debido a la promoción de las tarifas planas, permitiendo cada vez más variantes de comunicación a sus usuarios, e incrementando el ancho de banda a nivel internacional. El cometido final de la ingeniería de redes es que los usuarios puedan completar sus necesidades en forma eficiente y transparente, esto es, lograr una buena calidad de experiencia y sin que el mismo requiera adentrarse o intervenir en el mecanismo por el cual el servicio es exitoso.

Es debido a este contexto de incremento mundial del ancho de banda, y necesidad del compartir recursos (correos, fotos, documentos, películas, música, transacciones, contenidos multimedia en general y tantos más) que surge las Redes de Pares, o más conocidas como Redes “Peer to Peer ” (abreviadas con la sigla P2P), cuya filosofía de diseño contrasta a las clásicas técnicas previas cliente-servidor, de absorción de contenidos centralizados en un conjunto de servidores, y distribución de uno a todos.

Las redes P2P son redes virtuales desarrolladas a nivel de aplicación sobre la infraestructura IP (Internet Protocol). Los nodos de estas redes, denominados “pares”, ofrecen sus recursos (contenidos, ancho de banda y capacidad de procesamiento) a sus otros pares, básicamente debido a que comparten intereses en común. A medida que aumenta el número de pares integrantes de la red, también lo hacen los recursos globales de la misma, logrando una escalabilidad incomparable. Hoy en día, las redes de pares juegan un rol muy importante, dada su popularidad e impacto en Internet. La libre conducta de los pares en la red hace a P2P una herramienta atractiva para éstos. Por otra parte, impone grandes desafíos a la hora de diseñar su arquitectura y protocolos para compartir información.

En particular, las redes P2P de entrega de video en vivo tienen restricciones de diseño más complejas, fundamentalmente debido a que la cantidad de recursos en la red es dinámica (los pares bien pueden permanecer conectados en la red tan sólo unos minutos), sumado a las urgencias de tiempo real que suponen la transmisión en vivo. El desconocimiento del instante de ingreso y egreso de los pares a la red, como también el requisito de identificar otra entidad que aloje el contenido o porción del mismo solicitado, son cuestiones reiteradas para el diseño exitoso de protocolos de cooperación.

Las redes P2P poseen un impacto social y económico muy importante, que va mas allá de los

que las utilizan. Una cualidad de estas redes es la variedad de concepciones de anonimato. Esto es frecuentemente una ventaja para los usuarios de la red, pero un gran daño para el negocio de la industria discográfica y de derechos de autor. Los clientes comúnmente descargan contenidos libremente. Las grandes compañías RIAA (Asociación de la Industria Discográfica de Estados Unidos), la Sociedad General de Autores y Editores (SGAE) de España y la MPA (Motion Picture Association), entre muchas otras, han derivado incluso a juicios a usuarios que han llevado a disposición grandes cantidades de contenidos, infringiendo según los primeros, los derechos de autor. ¿Quiénes son los responsables? ¿Está bien que los usuarios obtengan gratuitamente los contenidos? Este aspecto controversial del uso de la tecnología P2P ha tenido un gran impacto social en la difusión de la información y ha colaborado para que los grandes proveedores de contenidos abran nuevos canales y modelos de negocios, más atractivos y favorables para el usuario final. Otro también es el negocio referente a los enlaces internacionales. Se ha estimado que el tráfico P2P ocupa cerca de un 70 por ciento del que se tiene a disposición en estos enlaces [28]. Esto repercute seriamente en el negocio de los Proveedores del Servicio de Internet (ISP, por sus siglas en inglés). Diversas técnicas de estos proveedores se han propuesto, con el fin de reducir, o al menos controlar este tráfico. Estas van desde el extremo de almacenar y descartar completamente tráfico P2P, a otras menos intrusivas, como redirigir este tráfico procurando caminos cortos (LiteLoad [49] y P4P [56, 90]), a aplicar servicios diferenciados (“shaping” de paquetes [24]), otorgando prioridad a aplicaciones que requieren “velocidad en sus transacciones”, según importancia e intereses del ISP. Los temas antes mencionados impulsan a un gran esfuerzo de científicos e investigadores. Existe una vasta variedad de artículos publicados referentes a diseños de protocolos de cooperación y arquitectura de redes P2P [25, 73, 94], como también del equilibrio entre los intereses del proveedor y los usuarios. Esta tesis se concentra en las redes P2P dirigidas a difundir video en tiempo real, cuya finalidad es lograr la mejor calidad de experiencia de los usuarios.

El contenido de esta tesis se presenta de la siguiente forma. Los Capítulos 1 y 2 muestran un marco teórico de las redes P2P y la cooperación entre pares. Los Capítulos 3, 4 y 5 contienen las principales contribuciones de esta tesis.

Más específicamente, el Capítulo 1 refiere a las medidas de calidad del video. Se resaltan aquí las técnicas de encuestado a los clientes, por estar más estrechas con la calidad de experiencia, y una aproximación a su aplicabilidad en tiempo real mediante la metodología pseudosubjetiva conocida como PSQA (Pseudo-Subjective Quality Assessment [74, 75]). Vamos a revisar distintos parámetros que impactan en la calidad de video. Particularmente en el video en vivo veremos que se destacan la continuidad en la reproducción, y la latencia inicial o “tiempo de buffering”. También se hace una revisión de distintas arquitecturas de redes de entrega de video (VDN del inglés), atendiendo sus diferencias en la distribución del contenido.

El Capítulo 2 comienza con una revisión de distintas redes P2P. Se va a destacar a continuación BitTorrent [31, 34]¹ famosa por su influencia en el diseño de protocolos de cooperación de otras redes P2P. Si bien BitTorrent muestra ser escalable (entendiendo escalabilidad como

¹<http://www.bittorrent.org>

la capacidad de admitir una cantidad incremental de usuarios, manteniendo la calidad de entrega de contenidos), está diseñada originalmente para la descarga (download) de contenidos, pero no adapta satisfactoriamente para redes de streaming en tiempo real. Posteriormente se introduce GoalBit [19]², que es la primera Red P2P de streaming de video en tiempo real de código abierto de gran difusión en Internet. Esta red mantiene la misma filosofía de diseño de BitTorrent, extendiendo su éxito a redes de streaming de video. Sin embargo, son necesarias diversas adaptaciones e innovaciones, que serán aquí explicadas a rasgos generales.

El Capítulo 3 se concentra en la cooperación entre pares a nivel de buffer, lo que se conoce como estrategia de selección de piezas a descargar. Veremos que a los efectos de entregar el video en tiempo real es impetuoso fragmentarlo en piezas (elemento imprescindible: básicamente es una manera de lidiar con las pérdidas de la red y adaptación a protocolos de transporte), que cada par debe conseguir para reproducirlo correctamente. Se introduce el modelo matemático simple de [97], el cual se concentra en los mecanismos de cooperación, atendiendo los parámetros de mayor impacto, que son la continuidad en la reproducción y su latencia. Veremos dos clásicas técnicas de cooperación, denominadas Rarest First (el primero más raro) y Greedy (la técnica “golosa”). Se agrega un análisis del desempeño de ambas, y una estrategia Mixta, que combina las mejores características de las clásicas técnicas.

En el Capítulo 4 se muestran las principales contribuciones de esta tesis. Basado en el modelo matemático detallado en el Capítulo 3, se propone una nueva familia de estrategias de selección de piezas. Esta posee interesantes propiedades, y riqueza en número y calidad suficiente para tener gran flexibilidad en la búsqueda de mejores estrategias. Además, se introduce un procedimiento sistemático de mejora de estrategias basado en una heurística de búsqueda local. También se estudian distintas medidas de optimalidad de estas estrategias considerando parámetros de impacto del video, y una estrecha relación de éstas cuando los usuarios idealmente poseen un tamaño de almacenamiento ilimitado, hipótesis no muy restrictiva, dado el incremental desarrollo de capacidad de disco duro de las computadoras personales. Finalmente se define un Problema de Optimización Combinatoria cuyo objetivo es capturar la calidad del video, y un enfoque de resolución basado en una traducción de una variante asimétrica del conocido problema TSP (Problema del Vendedor Ambulante [55, 87]). A partir de una metaheurística basada en Colonia de Hormigas (ACO [22, 38, 40, 41]) es que se resuelve este último, obteniendo así nuevas técnicas de cooperación entre pares para lograr altos desempeños en el video.

Por último, el Capítulo 5 estudia la emulación de la red Goalbit, cuya finalidad es obtener mejoras en la red real. La heterogeneidad y dinamismo de los pares son elementos no capturados en el anterior capítulo. El principal desafío es extender los anteriores resultados a esta red real, logrando escalabilidad. Se realizará aquí una comparación entre clásicas estrategias de selección de piezas, en relación con nuevas propuestas fruto del estudio teórico de los anteriores capítulos.

²<http://goalbit.sourceforge.net>

0.1. Publicaciones

De esta investigación se obtuvieron como producto las siguientes publicaciones ³:

- 24/5/2009 “*GoalBit: The First Free and Open Source Peer-to-Peer Streaming Network*”. Artículo aceptado en el 5th IFIP/ACM LANC 2009 (Latin-American Networking Conference 2009) para su publicación y presentación. El artículo fue presentado el 24 de setiembre, en la ciudad de Pelotas, Brasil, y se publicará en la biblioteca digital de ACM.
- 15/6/2009 “*Systematic Procedure for Improving Continuity and Latency on a P2P Streaming Protocol*”. Artículo aceptado en IEEE LatinCom 2009 (IEEE Latin-American Conference on Communications 2009) para su publicación y presentación. Fue presentado por el co-autor Lic. Darío Padula el 10 de setiembre en Medellín, Colombia. El artículo se publicará en IEEEXplore.
- 2/7/2009 “*A COP for Cooperation in a P2P Streaming Protocol*”. Artículo aceptado en ICUMT 2009 (International Conference in Ultra Modern Telecommunications 2009), para su publicación y presentación. El 13 de octubre en San Petesburgo, Rusia, será la ponencia de presentación del trabajo por parte del tesista. Este artículo se publicará en IEEEXplore.
- 16/9/2009 “*Estrategia de Selección de Piezas Optima para GoalBit: un sistema de streaming P2P basado en BitTorrent*”. Artículo aprobado para su presentación en el Segundo Congreso de Matemática Aplicada Computacional e Industrial (MACI 2009) en Rosario, Argentina, entre el 14 y 16 de diciembre.

³El equipo de investigación está compuesto por los siguientes miembros: María Elisa Bertinat, Daniel de Vera, Darío Padula, Franco Robledo, Pablo Rodríguez Bocca, Pablo Gabriel Romero y Gerardo Rubino

Parte II

MARCO TEÓRICO

Capítulo 1

Distribución y Evaluación del Video

1.1. Introducción

La Revolución Digital ha cobrado un intenso dinamismo a partir de la década de los 90'. La invención de la World-Wide-Web por el inglés Tim Berners en 1989 [15], permitió más adelante traducir el arduo manejo del lenguaje binario por las cómodas interfaces gráficas y comunicación multimedia. Tal fue el impacto en las comunicaciones, que en 1991 permitió el alcance de Internet al público, y tan sólo 4 años después, se lograron 50 millones de usuarios, cifra que la radio tardó 38 años en alcanzarla, desde su origen.

Es en este contexto que nace a mediados de los 90' la comunicación de video digital sobre redes de paquetes. Básicamente, esta consiste en una secuencia de cuadros que se despliegan en pantalla a más de 20 unidades por segundo, y bajo correcta operación logra un efecto de continuidad temporal al ojo humano. Por su parte, existen diversos desafíos para lograr la entrega de contenidos de video en Internet, como lo es la pérdida de paquetes (que contienen cuadros), la variabilidad del ancho de banda con el tiempo, el retardo y la fluctuación, entre muchas otras. Las estrategias de captura, manipulación, distribución y almacenamiento del video para lidiar con estos inconvenientes, ofrece un amplio ambiente de desarrollo en la investigación científica. Un elemento central para lograr altas calidades de video, es disponer de procedimientos de evaluación del desempeño del sistema. Aquí surge la pregunta: ¿cómo medir la calidad del video?

En este capítulo revisaremos en primera instancia las múltiples maneras con las que se concibe el video en redes de paquetes, enfatizando los mecanismos de su distribución. A continuación exploraremos las dos clásicas propuestas de evaluación de video, a saber: objetiva y subjetiva. Veremos que existe una nueva técnica híbrida de carácter pseudosubjetivo denominada PSQA (Pseudo Subjective Quality Assessment), más compatible con la medición de calidad de video en tiempo real, fundamental en el streaming. Por último, veremos los distintos parámetros de impacto en la calidad del video, atendiendo principalmente la aplicación de streaming de video, que concierne a la entrega y reproducción simultánea del video sobre redes de paquetes. Se destacarán la continuidad en la reproducción y la latencia, parámetros que se toman en cuenta para los siguientes capítulos de esta tesis. Estas secciones son un extracto de

lo presentado en [74]. Se recomienda su lectura por mayor profundidad.

1.2. Distintas concepciones del video digital

La calidad del video digital puede variar drásticamente en función de su técnica de compresión, distribución y medio físico de la entrega, entre muchas otras. En el caso de redes de paquetes, el mecanismo de entrega refiere a descarga (aplicación de “download”), en la que el usuario descarga todo el contenido de video y posteriormente lo reproduce en cualquier ocasión, o bien streaming, en el que el video es reproducido al mismo tiempo que se distribuye en la red.

El número de participantes de la comunicación varía según la aplicación. La comunicación de video puede ser 1 a 1 como ocurre en una videollamada o video bajo demanda, uno a varios (videoconferencias y juegos con múltiples participantes) o uno a todos (como la difusión televisiva). También se puede clasificar según la interactividad y bidireccionalidad. Por ejemplo, la videoconferencia es interactiva y bidireccional, mientras que la difusión televisiva (Broadcast TV) es unidireccional y no interactiva. Las limitaciones específicas de la aplicación particular afectan directamente el diseño del sistema.

1.3. Redes de Entrega de Contenidos (VDN)

Una VDN (Video Delivery Network) es un sistema que entrega el video en forma transparente al usuario. El área de VDN está creciendo actualmente, debido al crecimiento del ancho de banda en las redes de acceso (tanto en Internet como en sistemas celulares, redes IP privadas y otras) y debido al nuevo modelo de negocios propuesto por los productores de video y las nuevas opciones de hardware receptores de streaming de video. Desde el punto de vista de la red, las VDN presentan diversas arquitecturas, y retos tecnológicos particulares a cada una. A continuación mencionaremos algunas de las redes de entrega de contenidos más exitosas, como lo son la televisión por cable y satelital, Televisión por Internet, P2PTV, IPTV y TV en móviles.

1.3.1. Televisión por Internet

En Internet la mayoría de las VDN tienen una estructura tradicional de una red de entrega de contenidos (CDN) [33, 88], donde un conjunto de nodos absorben toda la carga, es decir, concentran la tarea de distribuir el contenido a sus clientes. Este es precisamente el caso de YouTube [93], msnTV [63], jumpTV [53], myTVPal [64], etc., todos trabajando con contenidos específicos de video.

1.3.2. P2PTV

Otro método popular de entrega de video en Internet hoy en día consiste en aprovechar la capacidad de los clientes, cooperando con el servidor para realizar la distribución del contenido, a través de los sistemas P2P (“peer to peer”). Actualmente, las redes P2P más famosas son

iMP [13](de la BBC de Londres), PPlive [70] y TVUNetwork [84], PPStream [71], SopCast [81] y TVAnts [83]. Estas son redes virtuales desarrolladas en la capa de aplicación sobre la infraestructura de Internet. Los nodos en la red, denominados pares, ofrecen sus recursos (ancho de banda, procesamiento, capacidad de almacenamiento) a otros pares, básicamente debido a que éstos comparten intereses en común. Como consecuencia, a medida que el número de pares en la red incrementa también lo hacen los recursos globales de la red P2P, permitiendo la escalabilidad de la red. El uso de una infraestructura P2P para la distribución de video aparenta ser una buena idea, debido a los altos requerimientos de ancho de banda de estas aplicaciones, elemento necesario para la distribución de streaming de video bajo demanda. No obstante, la transmisión de video streaming en vivo posee requerimientos muy exigentes, que se reflejan en una serie de problemas técnicos una vez decidida la distribución sobre una infraestructura P2P, a causa del gran dinamismo que presentan estas redes.

1.3.3. IPTV

IPTV es una arquitectura de red en la cual se entregan servicios mediante el protocolo IP sobre una infraestructura de red dedicada y cerrada. IPTV no es entonces difusión de televisión sobre Internet como P2PTV, sino que otro competidor en el mercado. Típicamente, IPTV es brindado por un operador en conjunto con voz (VoIP) y acceso a Internet, conjunto de servicios conocido como “Triple Play”. Dada la compleja arquitectura que poseen estas redes y la escasa experiencia de las tradicionales empresas de telecomunicaciones, en general las soluciones de IPTV provienen de un integrador. Algunos integradores de gran éxito en el mercado son Alcatel [5], Siemens [78] y Cisco [29].

1.3.4. MobileTV

Se utiliza este nombre para describir un servicio de entrega de video en dispositivos móviles. Este servicio normalmente lo ofrece el operador de la red celular. Una diferencia con los sistemas anteriores es que MobileTV tiene dos arquitecturas posibles en la red de acceso: puede utilizar la clásica red de acceso bidireccional celular (W-CDMA, CDMA-2000, GSM, GPRS, etc.), o una red de difusión digital dedicada (DVB-H, DMB, TDtv, ISDB-T, etc.). Se espera que la red celular sea utilizada para entrega uno a uno (VoD) y la red de difusión para la multidifusión televisiva en vivo. Las implementaciones tienden a cumplir con la arquitectura IMS (IP Multimedia Subsystem) [1], y, al igual que en el caso de IPTV, el mercado es dominado por los principales integradores.

1.3.5. TV cable y TV Satelital

La diferencia principal entre IPTV y TV Cable se encuentra en la etapa de distribución. Los proveedores de TV Cable ofrecen televisión a los usuarios mediante señales de radiofrecuencia transmitidas a través de cables coaxiales y redes de fibra óptica. Puede ser bien un sistema de una dirección o bidireccional. En el estandar DVB-C, el sistema de TV Cable utiliza un espectro con más de 100 portadoras de 6 MHz por canal, típicamente usados para transportar de 7 a 12 canales digitales (MPEG-2 requiere flujos de 3 a 5 Mbps). La modulación utilizada

es Modulación por Amplitud en Cuadratura (QAM) con 64 o 256 símbolos (64-QAM o 256-QAM). La televisión digital satelital por su parte utiliza un satélite para distribuir las señales a los usuarios. Típicamente utiliza Modulación por Fase en Cuadratura (QPSK) y los estándares MPEG-2 o MPEG-4 AVC y DVB-S (o DVB-S2).

1.4. Calidad del video

Un desafío común a cualquier VDN es asegurar una calidad mínima de video esperada por los usuarios finales del servicio. La calidad de experiencia (QoE) es básicamente una medida subjetiva del desempeño del servicio de extremo a extremo, desde el punto de vista de los usuarios. Como tal, es un indicador de la eficacia del sistema: cuan bien se logran sus cometidos.

1.4.1. Medidas objetivas y subjetivas

Históricamente existen 2 técnicas de evaluación de la calidad de video, a saber: objetivas y subjetivas.

Las primeras hacen uso de algoritmos y fórmulas que miden la calidad automáticamente, de modo cuantitativo y repetible. Definen una medida de la calidad del video distorsionado en base a una comparación con el original. En una red de paquetes frecuentemente toman como parámetros de entrada la tasa de bits (BR), tasa de pérdidas (LR), retardo y fluctuación, nivel de cuantización en la compresión, nivel de luminancia de la señal de video u otros. Algunos ejemplos de medida objetiva son PSNR (Peak Signal-to-Noise Ratio [7]), VQM (Video Quality Metric [69]), MPQM (Moving Picture Quality Metric) y CMPQM (Color MPQM [12]) y NVFM (Normalization Video Fidelity Metric [85]). Varían en la complejidad, como también en el nivel de correlación con la calidad subjetiva. PSNR por ejemplo es muy sencillo (tan sólo computa una suma de las diferencias entre niveles de luminancia a nivel de píxeles), pero está lejos de medir calidad de manera similar a los humanos. MPQM y su extensión con agregado de color CMPQM presentan mayor complejidad, y correlacionan bien con medidas subjetivas, al menos en escenarios particulares (con BR altos).

Por otra parte, las técnicas subjetivas definen la calidad en base a la opinión de múltiples usuarios, que juzgan la calidad de muestras de video distorsionadas. Es la mejor manera de determinar la calidad de experiencia de los usuarios, pues el video está pensado para seres humanos, y son ellos quienes tienen la última palabra. Sin embargo, presenta serias dificultades de aplicación para la evaluación del video streaming. La primera es que por definición no son posibles de realizar en tiempo real. La segunda es que su realización requiere disponer de recursos humanos, y consume mucho tiempo y esfuerzo. Existen medidas subjetivas tanto cualitativas como cuantitativas. Las cualitativas no se trasladan bien a escalas numéricas, aunque pueden ser útiles a la hora de negociar un contrato de servicio de entrega de video. En las medidas subjetivas cuantitativas, un grupo de personas juzgan la calidad del video con una puntuación numérica, y son más apropiadas para la medición de calidad de experiencia. La medida se brinda comúnmente a partir de un promediado de puntuaciones, conocido como MOS (Mean Opinion Score). El estándar ITU-R BT.500-11 detalla métodos para efectuar medidas subjetivas cuantitativas de calidad [52]. Las variantes van en el tiempo de toma de video, cantidad de

Cuadro 1.1: Escala de evaluación en DCR: la distorsión se caracteriza mediante las 5 opciones (original en inglés: imperceptible (5), perceptible, pero no molesta (4), levemente molesta (3), molesta (2) o muy molesta (1).

Puntuación	Descripción
5	Imperceptible
4	Imperceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

muestras, si se envían junto con el video sin distorsionar o no y conocimiento sobre video de los miembros del panel. A continuación mencionaremos una técnica subjetiva y cuantitativa de evaluación de video muy utilizada.

1.4.2. DCR (Degraded Category Rating)

El método subjetivo de evaluación DCR es también conocido como DSIS (Double Stimulus Impairment Scale), y se especifica en el estándar ITU-R BT.500-11. Las muestras de video se transmiten de a pares: en primera instancia el video original e inmediatamente después (no más de 10 segundos de espera) el video distorsionado, el cual debe ser puntuado por el panel según una escala del 1 al 5 que se muestra en el Cuadro 1.1. El tiempo total de una sesión para cada miembro del panel no debe superar la media hora para no agregar el efecto del cansancio, elemento a ser tomado en cuenta para determinar los días necesarios para tener los resultados finales. El estándar especifica el tiempo máximo por sesión, los segundos de espaciado entre el fin de la transmisión del video original y el distorsionado, y variantes de la metodología de evaluación (por ejemplo, enviar repetidas veces la secuencia de video original y la distorsionada).

A efectos prácticos, el resultado final va a ser un promedio o MOS para cada una de las configuraciones de video distorsionadas. Vale destacar que frecuentemente existe un proceso intermedio para eliminar “malas muestras”, consideradas como tales cuando se apartan de la media de opiniones (en el estándar ITU-R BT.500-11 se muestra una manera posible de filtrado estadístico en base a pruebas de normalidad y estimaciones de la desviación estándar). El MOS se vuelve a medir descartando estas últimas muestras.

1.4.3. Críticas a las Técnicas Históricas

Se observa que a los efectos de medir la calidad de experiencia del video en tiempo real no son aptas las estrategias clásicas de evaluación objetiva ni subjetiva. Las dos críticas a las objetivas son la baja correlación con la calidad percibida con los usuarios, e imposibilidad en la mayoría de las técnicas de obtener medidas de calidad en tiempo real, debido a la necesidad de poseer ambas muestras de video a la vez en un mismo extremo físico de la red.

Por otro lado, las técnicas subjetivas son el indicador de la calidad de experiencia de los usuarios. Sin embargo, por definición no son posibles de realizar en tiempo real, y además consumen tiempo y recurso humano, no siendo posible realizarlas con alta frecuencia.

Se propone entonces una nueva técnica pseudosubjetiva de evaluación de calidad de video, que no requiere las muestras distorsionada y original del video, y además es económica y fácil de automatizar, siendo posiblemente aplicable en tiempo real.

1.5. Otro enfoque: PSQA

PSQA (Pseudo Subjective Quality Assessment [61, 74]) es la técnica que permite cumplir con nuestros objetivos de medición de calidad de video streaming. La idea básica consiste en obtener en primera instancia una medida subjetiva de la calidad del video para una cantidad finita de configuraciones distorsionadas. Luego extrapolar esta función discreta (usualmente entrenando una red neuronal artificial (ANN)[65] o aleatoria (RNN) [45]) con otra continua, en función de los parámetros de impacto.

La etapa final es la evaluación pseudosubjetiva propiamente dicha, y sencillamente evalúa la función de calidad dados los parámetros de impacto del video distorsionado, que se deben medir.

Específicamente, el procedimiento para efectuar PSQA consiste en los siguientes pasos:

- 1- Determinar un conjunto de parámetros de impacto en la calidad de video.
- 2- Construir un escenario de pruebas con el que sea posible transmitir secuencias de video distorsionadas controladamente, eligiendo los valores de los parámetros de impacto antes seleccionados.
- 3- Elegir una cantidad razonable de configuraciones de video distorsionadas y un panel humano, para realizar una encuesta subjetiva.
- 4- Eliminar muestras malas, que son aquellas que difieren significativamente de la opinión media.
- 5- Calcular el MOS en base a las muestras buenas. Estos valores determinan para cada configuración distorsionada, la medida de calidad. Obsérvese que es tan sólo un conjunto finito de valores. Deseamos medir la calidad variando los parámetros en un continuo.
- 6- Entrenar una red neuronal aleatoria. Esta red tiene como entrada los parámetros de impacto normalizados, y como salida un número, que debe ser similar al MOS (tanto para las muestras de entrenamiento como las de prueba).
- 7- Efectuar la medida pseudosubjetiva propiamente dicha de la calidad mientras fluye en una red de paquetes un streaming de video, midiendo los parámetros de impacto y evaluando la función antes obtenida.

Las Figuras 1.1 y 1.2 ilustran la etapa previa a la medición (o de “entrenamiento” de la red neuronal) y la de utilización propiamente dicha, respectivamente. Nótese que una vez realizado

el entrenamiento, si los parámetros de impacto siguen capturando la esencia de la calidad de experiencia del video, la misma red entrenada puede seguir siendo utilizada, y el proceso de encuestado y obtención de MOS lo realizamos tan sólo una vez.

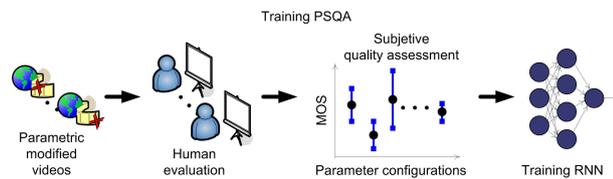


Figura 1.1: Etapa de entrenamiento de PSQA.

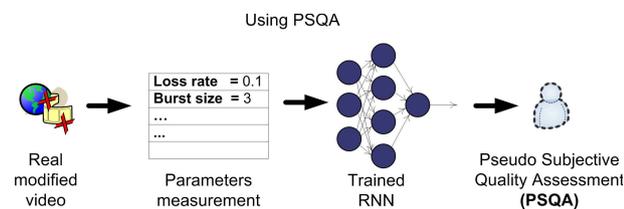


Figura 1.2: Aplicación de PSQA.

1.6. Parámetros de Impacto

El primer paso, previo al encuestado de PSQA, es definir los parámetros a medir que impactan en la calidad de video. Para que la encuesta sea factible, se debe elegir una cantidad limitada de estos parámetros. Además los mismos deben ser fáciles de medir y distorsionar, y lograr capturar la esencia de la calidad subjetiva: deben ser los más impactantes en la calidad. Estos parámetros pueden ser parte del ambiente o escena visual, ser característicos de la fuente, propios de la red o a nivel de aplicación.

1.6.1. Parámetros de ambiente

Son por ejemplo el nivel de iluminación, ruido de fondo, parlantes y monitor considerado para el despliegue, entre otros. Si bien tienen un impacto directo en la calidad, frecuentemente el proveedor del servicio no puede controlarlos, sino que están más bien a manos del usuario. De aquí que estos parámetros no son considerados para desarrollar encuestas y posterior aplicación de PSQA.

1.6.2. Parámetros de la fuente

El medio con el que se genera el video tiene un impacto obvio en la calidad global percibida. Entre ellos están los niveles de luminancia y sonido, contraste y tamaño de cuadros.

Los parámetros de compresión son los más importantes de la fuente: codec utilizado, tasa de muestreo, bits por muestra, tasa de bits, tasa y tipo de cuadros, etc. El emisor ante conocimiento de las características del canal puede efectuar distintas estrategias de transmisión, como lo son el entrelazado en la imagen, corrección de errores hacia adelante (FEC) y codificación.

1.6.3. Parámetros de distribución

Las medidas de la Calidad del Servicio (QoS) a nivel de red son generalmente los principales componentes en el diseño y gestión de la misma. Típicamente incluyen pérdidas de paquetes, retardo, fluctuación y ancho de banda. Desafortunadamente, no es claro en qué medida impactan estos parámetros en la QoE del video percibido. Más aún, las distintas aplicaciones son sensibles o tolerantes a estos distintos parámetros. Por ejemplo, si la aplicación requiere interactividad (como una videollamada), el sistema no tolera altos retardos ni fluctuaciones, y éstos tienen un rol central.

Internet es una red de “mejor esfuerzo”, lo que significa que no hay garantías de calidad de servicio a los usuarios finales. No obstante, es posible aplicar ciertas técnicas para contrarrestar los efectos de la red en la calidad final. Algunos de ellos aplican gestión activa de colas (AQM), como la detección temprana aleatoria (RED [4, 23]) que permite descartar paquetes aleatoriamente y con probabilidad de descarte incremental con el tamaño de paquetes a ser servidos, hecho que promueve a evitar congestiones en la red, y mejorar así la calidad del servicio.

Los protocolos a nivel del transporte, como lo son RTP (Real-time Transport Protocol [76]) y protocolos de señalización de la aplicación (SIP y H.323, entre muchos otros) son importantes también en la calidad, pues permiten agregar retransmisión, información de sincronización y algoritmos de control de congestión.

Las redes P2P deben ser robustas ante la variabilidad de los recursos. Para encarar el problema de desconexiones y por ende estas fluctuaciones de recursos, las técnicas más utilizadas consisten en agregar redundancia. De esta manera, la información está disponible o dispersa entre los nodos de la red, y el esfuerzo siguiente radica en métodos para compartir información entre terminales.

1.7. Latencia y Continuidad en video streaming

Estos dos parámetros de video son los más estudiados en el caso particular del streaming [10, 43, 50, 60, 77]. Estudiaremos en esta sección la importancia de estos parámetros, y el motivo por el cual se destacan en el contexto del video streaming.

1.7.1. Continuidad del Video

Es bien conocido de trabajos previos en PSQA que el proceso de pérdidas genera el mayor impacto en la calidad de video [61]. La continuidad de reproducción del video puede medirse de diferentes maneras, en todas contabilizando pérdidas de información de video. Normalmente, el contenido de video es fraccionado en piezas, que básicamente contienen algunos cuadros

de imagen que componen la secuencia. No obstante, a efectos de ser transmitido por una red, este video es cortado en menor escala, en elementos denominados paquetes. De aquí que es posible medir continuidad visto desde la red contabilizando la proporción de paquetes arribados en recepción, o desde la aplicación, estudiando por ejemplo la relación de cuadros arribados y los necesarios para completar el video entero, o la cantidad de piezas perdidas con respecto a las que componen toda la secuencia de video. En [74] se realiza por primera vez un estudio detallado de pérdidas a nivel de cuadros, aplicando el modelo de Gilbert simplificado [62] para atender dos parámetros orientados a pérdidas: la tasa de pérdida de bits o bit rate, y el largo medio de pérdidas en ráfagas o MLBS (Mean Loss Burst Size). Allí se muestra que el estudio de pérdidas a nivel de cuadros posee mayor relación con la calidad subjetiva que los modelos de pérdidas a nivel de paquetes. Otros trabajos se concentran en pérdidas de paquetes [72], [44].

En esta tesis nos concentraremos en el estudio de diversas técnicas de cooperación entre pares a nivel de aplicación. Puesto que los pares a nivel macroscópico (y suficiente para el entendimiento de la cooperación) intercambian piezas de video, mediremos la continuidad de reproducción en base a pérdidas de piezas. En el Capítulo 3 veremos un modelo matemático de cooperación entre pares, y una definición formal de continuidad, en términos probabilísticos de la obtención de la próxima pieza a ser reproducida.

1.7.2. Latencia inicial del Video

La latencia inicial o “tiempo de buffering” se define como el tiempo que transcurre desde que un par se conecta a la red hasta que comienza a reproducir el video deseado. Si bien por razones prácticas PSQA no contempla este parámetro, es sin embargo de gran impacto en el desempeño del streaming, básicamente debido a que un usuario normalmente no acepta esperar minutos para iniciar la reproducción a partir de su conexión. En el diseño de redes de streaming de video es natural enfrentarse al desafío de lograr una alta calidad de reproducción (continuidad) sujetos a reducidos tiempos de buffering. Intuitivamente, esta latencia inicial permite preparar, acondicionar un buffer con abundantes piezas de video mediante una técnica de cooperación, de modo de asegurar la continuidad al inicio. Es razonable, y frecuente a partir de lo anterior, estudiar el compromiso entre el tiempo de latencia inicial y la continuidad. Ejemplos de trabajos en esta dirección son [60, 77]. En esta tesis se caracteriza la calidad del video streaming a partir de estos dos parámetros. Se profundiza en este compromiso en los Capítulos 4 y 5.

1.8. Conclusiones

En este capítulo hemos revisado brevemente las distintas formas que un usuario dispone de obtener el video. Se ha destacado la importancia de juzgar la calidad del mismo como elemento fundamental para la concepción de la eficacia de una red de entrega de contenidos, en base a la calidad de experiencia. La técnica de evaluación de video pseudosubjetiva, PSQA, se muestra como la mejor propuesta para medir la calidad de video en tiempo real, en contraste con las técnicas objetiva y subjetiva. Existen diversos factores que afectan la calidad de video. Trabajos de PSQA dejan en evidencia que los procesos de pérdida son los de mayor impacto en el video.

El parámetro en esta tesis que toma en cuenta las pérdidas es la continuidad de reproducción de video, en base a la obtención de piezas. Estas van a ser la mínima unidad de intercambio entre pares en los modelos matemáticos aquí presentes. Por otra parte existe un compromiso muy interesante en redes de streaming de video entre la continuidad y la latencia inicial. Cuando un usuario dispone de opciones para contratar un servicio, es fundamental que esta latencia o tiempo de buffering sea lo más reducida posible. El compromiso entre latencia y continuidad es un tema central en este trabajo.

Capítulo 2

BitTorrent, GoalBit y Video en Vivo

2.1. Introducción

En este capítulo revisaremos las particularidades de la cooperación entre pares en el protocolo BitTorrent. El mismo ha tenido gran éxito gracias a su escalabilidad y robustez a los efectos de realizar descargas de contenidos. No así para aplicaciones que requieren tiempo real como el streaming de video.

Introduciremos la plataforma GoalBit ¹, que es la primera red de difusión de video streaming de código libre utilizada a gran escala en Internet. La misma adapta la filosofía de BitTorrent, procurando extender ese éxito a difusión de contenidos en tiempo real. Veremos aquí importantes contrastes con la misma. Entre ellos se resaltan la arquitectura en jerarquías, nuevos conceptos a nivel de buffer que hacen prioridad a la toma de piezas cercanas a la línea de reproducción y la política y participación del tracker de GoalBit, entre otras. Aquí se presenta una descripción resumida de GoalBit. Se sugiere al lector la referencia [19] por más detalles.

2.2. BitTorrent

2.2.1. Introducción

El protocolo BitTorrent fue desarrollado por el programador Bram Cohen en el año 2001 [20]. Tiene un fuerte sustento con la teoría de juegos y de la información. La clave de su escalabilidad es la cooperación basada en incentivos: “dar para recibir” [31]. Su popularidad también se ve beneficiada por la simplicidad del uso de su cliente, y por ser uno de los pocos sistemas P2P de protocolo abierto.

Actualmente BitTorrent mantiene su gran popularidad, logrando una porción importante dentro del tráfico P2P de Internet [28]. Para transmitir los contenidos se procede cortando los mismos en cientos o miles de piezas, y cada par mediante una estrategia de cooperación (como veremos, preguntando por piezas más raras) debe armar ese rompecabezas, a partir de lo entregado por otros pares. Un informe muestra que la mitad del tráfico de Internet es de BitTorrent [26]. Desde sus inicios ha sufrido transformaciones, y se presenta en distintas versiones.

¹<http://goalbit.sourceforge.net>

Veremos a continuación cómo percibe un cliente el servicio ofrecido por este protocolo, vía un cliente BitTorrent. Luego mencionaremos sus particularidades de cooperación y otras estrategias de comunicación utilizadas.

2.2.2. Descargar archivos con BitTorrent

Es muy fácil descargar contenidos con BitTorrent. Lo primero que se debe hacer es descargar un cliente BitTorrent en el disco duro del ordenador. A continuación navegar en la web hasta hallar un archivo con extensión .torrent. Con un simple click, basta con elegir el directorio donde realizar la descarga y el cliente se ocupa de descargar completamente el contenido. Una vez completo hay que cerrar manualmente el descargador. El usuario ya puede disfrutar del contenido, que se halla en su disco duro.

2.2.3. Arquitectura de BitTorrent

La red de BitTorrent consiste de un servidor web que aloja el contenido .torrent estático, pares que representan los usuarios de la red cuya finalidad es descargar los contenidos y un tracker, que es el único punto de coordinación entre los pares que comparten el mismo contenido. Este mantiene información actualizada del estado de los pares, y hace factible la comunicación entre ellos, brindando a cada nuevo usuario una lista de posibles con los cuales conectarse, denominada usualmente “swarm” (enjambre o nube de pares). Los archivos .torrent tienen un campo que permite ubicar al tracker. De esta manera se procede consultando al mismo, quien posteriormente brinda el swarm al par. Una vez que el cliente de un par inicia la descarga, comienza la consulta de piezas del contenido a sus otros vecinos. Este par puede cooperar desde que descarga tan sólo una pieza. Así es que BitTorrent aprovecha el ancho de banda de cada par lo más temprano posible, posibilitando una visión de ancho de banda global en la red, y favoreciendo la escalabilidad.

2.2.4. Estrategia de Selección de Piezas

2.2.4.1. Política de Rarest First

Para maximizar la disponibilidad de la información de la red, es deseable lograr que existan abundantes réplicas de cada una de las piezas del contenido. Si se considera azarosa la manera de comunicarse con un par, es equivalente a maximizar el valor esperado de la información (o la entropía) neta en la red. En este sentido, BitTorrent aplica la estrategia de consulta solicitando las piezas “más raras” de la red, técnica conocida del inglés como Rarest First. Es así que se tiende a uniformizar la distribución de probabilidades de solicitud de una nueva pieza. En [27] se muestra un modelo matemático de complejidad NP para decidir una estrategia de selección de piezas, y analiza la calidad de la estrategia Rarest First, clasificándola como subóptima pero con muy buen desempeño. En [97] se introduce Rarest First como una estrategia de selección que logra la escalabilidad, mas su latencia es muy pobre. El modelo matemático usado

en [97] es el que se elige profundizar en esta tesis, debido a su sencillez y la claridad con la que se captura la cooperación a nivel de buffer de la red. En el Capítulo 3 se detalla este modelo matemático y se ilustran distintas estrategias de selección. Ya en el Capítulo 4 se profundiza en la elección de distintas estrategias, con nuevos enfoques de búsqueda de las mismas.

2.2.5. Estrategia de Selección de Pares

2.2.5.1. Tit-for-tat

Las redes P2P sufren de parasitismo. Existen pares que sin cooperar con sus vecinos en la red descargan contenidos, muchas veces denominados “free riders” (sanguijuelas). Los modelos económicos introducen muy frecuentemente este concepto como un comportamiento racional de sus componentes participantes [68]. Algunos ejemplos cotidianos de free riding es el usuario de un transporte público que no paga su boleto. Otro puede ser el beneficio popular de la defensa militar. En el caso presente, refiere a pares que descargan sin subir piezas a la red. Es posible por lo tanto utilizar la teoría de juegos, que es un marco sólido para analizar el problema y tomar medidas al respecto. La estrategia “tit-for-tat” tiene su origen en el problema del prisionero iterado [36]. Se ha demostrado que es la estrategia óptima en este juego, dentro de todas las estrategias determinísticas. Su nombre proviene del inglés “tit for tat”, que significa “represalia equivalente”. Si un oponente coopera antes, la decisión será cooperar. En caso contrario si deserta, la decisión será desertar. Curiosamente, esta estrategia fue diseñada en 1984 en un concurso para hallar la mejor estrategia al problema del prisionero iterado por Anatol Rapaport [9], quien propuso una solución expresable en tan sólo 4 líneas en el lenguaje BASIC, ganando a todas las demás estrategias, muchas extensas y muy complejas. Ahora bien: ¿cómo se aplica “tit-for-tat” en una red P2P? Cada par se responsabiliza de procurar una alta tasa de bajada, descargando piezas de todos los pares que pueda. Cuando tiene la oportunidad de decidir con qué par va a cooperar se aplica una variante de tit-for-tat. Entonces, si un par no comparte su información, los pares con los que mantiene comunicación actuarán del mismo modo. Este es un importante incentivo para cooperar. Sin embargo, hay un problema aún no resuelto: ¿siendo un par nuevo de la red, cómo me es posible cooperar si aún no poseo ninguna pieza?

2.2.6. Optimistic Unchoking

De todo el swarm de pares conocidos por un par, sólo se intercambian piezas con unos pocos. Cada par decide mantener un intercambio de piezas exclusivamente con aquellos que le entregaron más piezas últimamente (típicamente 4 pares). Estos pares están en estado de desbloqueo (unchoke) para este par (todo el resto del swarm permanece bloqueado o choke). No es posible mantener un intercambio fluido con todo el swarm por limitantes ingenieriles. BitTorrent opera sobre TCP (Transmission Control Protocol [2, 3]). Ante el inicio de un gran número de conexiones TCP, la congestión en los cuellos de botella son un duro problema, que se puede reducir, o incluso evitar con bloqueos o con políticas de gestión de colas (AQM). Por otra parte, una vez que se desbloquean, se permite regularmente una conexión con pares no conectados al azar. Esto tiene el fin de conocer nuevas conexiones, a los efectos de lograr

posiblemente mejores resultados que los antes alcanzados por el par. A su vez, permite lidiar así con el problema de los pares que ingresan a la red, que no pueden cooperar. Esta técnica se aplica en BitTorrent, y se denomina “optimistic unchoking”. La política de choking es crítica para el buen desempeño de una red P2P.

2.2.7. Conclusiones Parciales

El protocolo BitTorrent muestra todo su potencial en la cooperación de pares. Introduce los conceptos de dar para recibir (sustentado y típico de la teoría de juegos), de pedir el primero más raro (para maximizar la entropía y por ende la disponibilidad de la red), y del desbloqueo optimista, para dar oportunidad a nuevos pares de establecer una comunicación y cambiar las perspectivas de pares bloqueados.

Si bien la política de solicitar la primera pieza más rara funciona para descarga de contenidos estáticos, no conforma los requerimientos de tiempo real, básicos en una red de streaming de video. Además, BitTorrent está pensado para descargar contenidos de tamaño fijo, pero no un flujo continuo e ilimitado. Surge naturalmente en estos tipos de redes, la necesidad de consultar por piezas más próximas a la línea de reproducción. Pero por otra parte solicitar siempre piezas más próximas a la línea de reproducción es una técnica golosa y no escalable. Se necesita aquí un delicado equilibrio, que reside en lograr una calidad de reproducción satisfactoria al ojo humano (no perder piezas de video), pero a su vez cuidando el tiempo de latencia inicial.

A continuación estudiaremos una plataforma real denominada GoalBit, que tiene como cometido la difusión de contenidos con exigencia de tiempo real, como lo es el video streaming. Aquí se mostrarán las diversas adaptaciones que deben aplicarse a BitTorrent para lograr tiempo real, y los desafíos inherentes al diseño de un protocolo de una plataforma real de distribución de video sobre redes P2P.

2.3. GoalBit

2.3.1. Introducción

GoalBit es el primer sistema de distribución de video streaming P2P usado en Internet, de fuente y protocolos abiertos. En este apartado se presentará la plataforma GoalBit, resaltándose dentro de sus desafíos de diseño la necesidad de tiempo real, el dinamismo de la red y la heterogeneidad de sus pares. Actualmente, GoalBit es utilizado por operadores y usuarios finales para recibir y enviar video en vivo. Veremos su arquitectura, sus protocolos y la técnica de cooperación, resaltando las principales diferencias con BitTorrent. Por detalles sobre la implementación de GoalBit se sugiere leer [47].

2.3.2. Streaming de Video P2P

Los nodos se conectan usualmente a la red tan sólo unos minutos [82]. Hoy en día, existen redes comerciales P2P a disposición. Dentro de las más exitosas se encuentran PPLive [70] superando los 200000 usuarios concurrentes a un mismo canal, SopCast [81] con más de 100000

usuarios, tal como reportan sus desarrolladores, CoolStreaming [32] con más de 25000 usuarios sobre un mismo canal, PPstream [71], TVAnts [83], y TVUnetwork [84], todas de código y protocolos propietarios.

PPLive es el software más popular de P2PTV, especialmente en Asia. Aplicando técnicas de ingeniería reversa, los trabajos [48] y [6] muestran que PPLive se basa en el protocolo BitTorrent, con una selección de canal a partir de un Webtracker, y transferencia de video a partir del uso de piezas. Utiliza dos formatos de video: Windows Media Video (VC-1) y Real Media Variable Bitrate (RMVB), con tasas que varían de 250 kbps a 400 kbps, utilizando el reproductor de video del usuario. La política de reproducción no es Rarest First, pues se debe cumplir con la restricción de tiempo real. Tampoco se utiliza la técnica de tit-for-tat. Cuando un usuario ingresa a la red, sus pares vecinos (del mismo swarm) le envían piezas forzosamente al primero para minimizar la latencia inicial en la reproducción. También a partir de ingeniería reversa se muestra que SopCast utiliza un protocolo propietario con características muy similares a PPLive (más detalles de SopCast se pueden encontrar en [77]).

CoolStreaming es el predecesor de PPLive y SopCast, ahora forzado a cerrar debido a derechos de autor. Sus autores publicaron su aparato de distribución [92, 95, 96]. El video se divide en piezas de igual tamaño. Una diferencia con respecto a sus sucesores es que el flujo de video se divide en 6 sub-flujos con una selección de piezas de video por turnos.

A los efectos de la generación del video en GoalBit, este se puede capturar con diversos dispositivos, como una tarjeta de video, una Webcam, un streaming HTTP/MMS/RTP, un archivo, y otras. Se puede codificar con diferentes formatos (MPEG2/4-AVC, VC-1, ACC, VORBIS, MP-GA, WMA, etc.). El empaquetamiento, definido mediante GoalBit Packetized Stream (GBPS), toma el flujo y genera piezas de tamaño fijo, y son luego distribuidas utilizando el Protocolo de Transporte GoalBit (GBTP) que se basa en BitTorrent. Finalmente, el cliente GoalBit reproduce las piezas obtenidas con GBTP. Para la reproducción se integran el Videolan Media Player al código fuente, por lo tanto el usuario no requiere un reproductor de video adicional.

2.3.3. La Plataforma GoalBit

Hay 4 tipos de nodos de la red: el broadcaster, los superpares, los pares y el tracker. Los broadcasters son los responsables de introducir el contenido en la plataforma e iniciar su distribución. Los superpares son pares con alto ancho de banda, y su principal rol es colaborar con la distribución inicial del contenido. Los pares son usuarios finales, que se conectan a algún canal para reproducir un contenido de video. El tracker tiene el mismo rol que en BitTorrent: tienen a su cargo la gestión de los pares del sistema. Usualmente, algunos servidores web alojan archivos .goalbit, y los usuarios los obtienen de allí. Un diagrama ilustrativo se puede ver en la Figura 2.1.

2.3.4. Protocolo de Transporte GoalBit

En GBTP, cada pieza de video es identificada por un número único, llamado "piece ID". Se utiliza para facilitar el intercambio de piezas entre pares. Dado que GBTP está pensado para

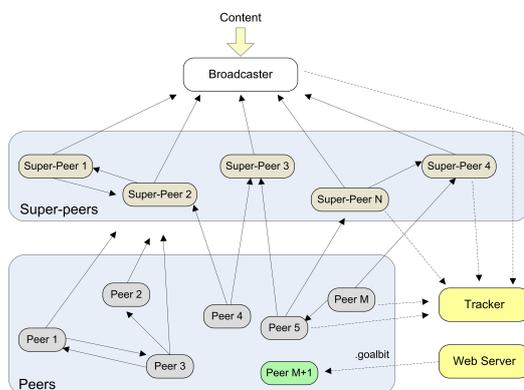


Figura 2.1: Arquitectura GoalBit. Los 4 componentes de la red son el broadcaster, los superpares, los pares y el tracker.

distribuir un flujo continuo y no archivos, el número de pieza es cíclico en el intervalo $[0, 2^{31}]$. Cada par posee una ventana deslizante que admite un rango de ID. Tiene un valor base que es el mínimo ID de la ventana, y un largo (que define el máximo valor de identificador de la ventana). Los pares sólo pueden descargar o compartir piezas que estén comprendidas en sus respectivas ventanas. Tienen un reproductor que consume de modo secuencial las piezas almacenadas en la ventana deslizante. El índice de ejecución se define como el siguiente ID a ser consumido por el reproductor.

Otros conceptos importantes usados en el protocolo son el buffer activo (active buffer) y el índice de buffer activo (ABI). El *active buffer* se define como la secuencia consecutiva de piezas que un par posee, comenzando del índice de ejecución. En otras palabras, es el video descargado que se va a reproducir sin interrupciones en un tiempo específico. Por último, el ABI es el mayor ID dentro del buffer activo. La Figura 2.2 ilustra estos conceptos.

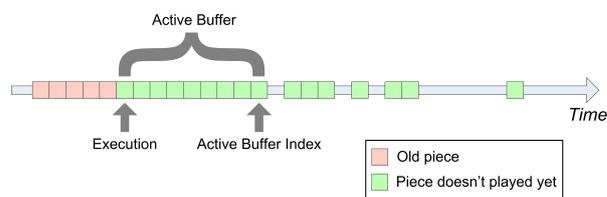


Figura 2.2: Buffer Activo e índice del Buffer Activo (ABI).

2.3.5. Información en el tracker

El tracker almacena una lista de todos los pares que se conectan a cada canal. Cada canal tiene un identificador denominado “info hash” como también información de los pares que están conectados. Por cada par se almacena información estática y dinámica. La información

estática consiste en la IP de los pares, el puerto que utilizan para sus nuevas conexiones, y el tipo de par, que puede ser broadcaster, par, superpar o broadcaster-superpar (este último utilizado en redes que no poseen superpares). Entre sus datos dinámicos se hallan el ABI, la QoE percibida y la hora de su último reporte de presencia.

2.3.6. Archivos .goalbit

Los archivos .goalbit son contenedores de canales. En ellos se halla la información necesaria para ejecutar el streaming desde un par (dirección del tracker, identificador de canal, etc.). Estos archivos son muy similares a los correspondientes a BitTorrent, exceptuando que los archivos GoalBit son textos planos estructurados con XML. Puesto que los archivos GoalBit pueden contener información sobre múltiples canales, es posible establecer un canal por defecto, a ser ejecutado cuando se abre el archivo.

2.3.7. Comunicación Par-Tracker

La comunicación par-tracker es iniciada por el par mediante un mensaje de anuncio (announce). Los Cuadros 2.1 y 2.2 muestran los campos dentro de una solicitud announce y las posibles respuestas del tracker, respectivamente. Inicialmente, los pares contactan al tracker de modo de obtener una lista de pares y un índice a partir del cual se inicia a preguntar por piezas. Normalmente, los pares envían un mensaje announce al tracker con $numwant = 55$ y $ABI = MAXPIECE + 1$. En la respuesta, el par obtiene una lista de pares con al menos tantos elementos como el número solicitado, el índice de comienzo y el tipo de par asignado. Puesto que el tracker necesita almacenar información del estado de cada par, debe existir una comunicación regular entre ellos para permitir la transferencia de esta información. En la comunicación periódica, el par envía un anuncio con $numwant = 0$, el índice de buffer actual ABI y la QoE medida por el par en el último período. En este caso, no hay una respuesta relevante por parte del tracker. En estos intercambios, el par puede solicitar más vecinos. Esta comunicación es regulada por el tracker, quien define el intervalo de tiempo mínimo entre dos anuncios seguidos. Según el tipo de par con el que se conecta el tracker, la respuesta contiene una lista de pares diferente. A los broadcasters directamente no se envía lista. Los superpares reciben una lista con broadcasters y otros superpares. Por último, los pares deben recibir una lista con muchos pares y pocos superpares. Hay muchas estrategias posibles para armar el swarm por parte del tracker. Hoy en día, existe una muy popular conocida como P4P (Participación Proactiva del Proveedor [35, 91]), que sugiere que la lista enviada a un par debe estar compuesta por pares geográficamente próximos al que inicia la conexión con la red. Además, es deseable que los pares de la lista estén descargando piezas temporalmente cercanas a aquellas que necesita el par que recibirá la lista (es decir, piezas próximas al ABI del receptor). GoalBit puede construir el swarm para cada usuario según el método tradicional de BitTorrent (aleatorio y uniforme), usando P4P [67, 90] o la variante menos difundida propuesta por ONO.

2.3.8. Comunicación Entre Pares

Una vez que el par recibe un índice inicial de ABI y lista de pares, puede iniciar la comunicación con los pares de su lista. Los intercambios de mensajes se realizan sobre el protocolo

Cuadro 2.1: Parámetros enviados por el par en un mensaje Announce al tracker.

Parámetro	Descripción
<i>protocol</i>	Versión del Protocol utilizada, actualmente "GoalBit-0.4".
<i>info_hash</i>	Identificador de canal (cadena SHA1 de 20 bytes).
<i>peer_id</i>	Identificador de Par. Cadena aleatoria de 20 bytes, generada por el par al inicio de la sesión.
<i>event</i>	Toma valores "started" o "stopped". Tanto en el inicio como en la comunicación periódica debe valer "started". Se debe enviar "stopped" solamente ante la desconexión del par.
<i>port</i>	Puerto de donde el par escucha conexiones entrantes.
<i>uploaded</i>	Bytes subidos desde el inicio de la ejecución.
<i>downloaded</i>	Bytes bajados desde el inicio de la ejecución.
<i>numwant</i>	Número de pares que el tracker debería retornar al par. Si vale 0, el tracker no retorna pares.
<i>ABI</i>	Valor de ABI actual del par (en el instante de envío). En el inicio (cuando aún no dispone de un ABI), se debe enviar el valor $MAX_PIECE_ID + 1$.
<i>peer_type</i>	Tipo de par: Broadcaster = 1, Superpar = 2, Par = 3, Broadcaster-superpar = 4.
<i>peer_subtype</i>	Subtipo de par (sólo usado en casos particulares).
<i>QoE</i>	Calidad de experiencia medida por el par.
<i>Compact</i>	Vale 1 cuando el par acepta una respuesta compacta (en la respuesta del tracker cada par se enviará como una cadena de 6 bytes, donde los primeros 4 bytes representan la IP del par y los últimos 2 su puerto).
<i>tracker_id</i>	Dentro de la respuesta del tracker es posible que incluya un parámetro llamado <i>tracker_id</i> . Si esto ocurre, en los siguientes mensajes de anuncio se debe retornar este parámetro al tracker.

Cuadro 2.2: Parámetros enviados por el tracker en respuesta a la solicitud de los pares.

Parámetro	Descripción
<i>interval</i>	Intervalo de tiempo(en segundos) que el par debe esperar antes de enviar un nuevo anuncio al tracker.
<i>tracker_id</i>	Este parámetro debe ser devuelto al tracker en los siguientes anuncios.
<i>broadcaster_num</i>	Cantidad de Broadcaster involucrados en el streaming.
<i>super-peer_num</i>	Cantidad de superpares involucrados.
<i>peer_num</i>	Número de par en el streaming.
<i>max_ABI</i>	Máximo ABI reportado por un seeder.
<i>peer_type</i>	Tipo de par asignado al par al que se responde.
<i>offset</i>	Índice inicial (desde donde el par debe iniciar la consulta de piezas). Este parámetro se utiliza solamente al inicio de la comunicación.
<i>peers</i>	Lista de pares.

TCP. Hay dos tipos de mensajes intercambiados. Los primeros se utilizan para informar sobre la situación actual de cada par, detallando las piezas que posee cada uno, como también los vecinos que permanecen conectados. Hay cinco tipos de mensajes utilizados para este primer intercambio de información, a saber: HANDSHAKE, BITFIELD, HAVE, WINDOWUPDATE y KEEPALIVE. El otro grupo de mensajes se utiliza para intercambiar piezas. Como veremos a continuación, el proceso de intercambio de piezas no es trivial, y se utilizan ocho tipos de mensajes: INTERESTED, NOT INTERESTED, CHOKE, UNCHOKE, REQUEST, CANCEL, PIECE y DON'T HAVE.

2.3.8.1. Mensajes de control entre pares

Inicialmente, el par envía mensajes de tipo HANDSHAKE (detalles en el Cuadro 2.3) a cada par de su lista, con el fin de establecer una conexión. Entre otras cosas, en estos mensajes se envía información tal como la clase de par y la ventana deslizante (su base y ancho). Esta información es utilizada por el par receptor para decidir desde ya si aceptar la comunicación o rechazarla (cuando las ventanas no se solapan es razonable esperar que la conexión se rechace, dado que no se van a poder beneficiar mutuamente, y así evitar el parasitismo). En caso de aceptación, el receptor debe enviar otro mensaje de HANDSHAKE con sus datos. Una vez que se establece la conexión, proceden intercambiando mensajes del tipo BITFIELD (ver Cuadro 2.4), que contiene la ID de aquellas piezas que le faltan al receptor y posee el transmisor (aquellas deseables por el último terminal). Con mensajes de este tipo es que cada par es informado

Cuadro 2.3: Definición del mensaje de “HANDSHAKE”. Este es el mensaje inicial en la comunicación entre 2 pares. Técnicamente, debe ser enviado por cada terminal cuando se efectúa una nueva conexión. El mensaje tiene 77 bytes de largo.

Campo	Tamaño (bytes)	Descripción	Valor
<i>pstrlen</i>	1	Aquí se brinda el largo <i>pstr</i> del campo	16
<i>pstr</i>	<i>pstrlen</i>	Identificador de Protocolo (cadena de caracteres)	Protocolo GoalBit
Reserved	8	Byte reservado por el protocolo	0
<i>info_hash</i>	20	Identificador de la transmisión. Este campo es un hash SHA1 de 20 bytes del identificador del canal	N/A
<i>peer_id</i>	20	Identificador del par. Cadena al azar de 20 bytes para la identificación de usuario	N/A
<i>peer_type</i>	4	Tipo de par: broadcaster = 1, superpar = 2, par = 3, broadcaster superpar = 4	N/A N/A
<i>offset</i>	4	Base de la ventana deslizante	N/A
<i>length</i>	4	Largo de la ventana deslizante	N/A

de las piezas que el resto de sus vecinos le puede brindar. Para tener esta información actualizada, cuando un par recibe una nueva pieza debe enviar un mensaje HAVE (ver Cuadro 2.5) a todos los pares que no posean ni hayan reproducido la misma, y a la vez que puedan recibirla (que su ventana contenga el índice de esta pieza). Los pares a su vez van corriendo la ventana a medida que avanza el tiempo. Esto debe ser informado a todos los pares que permanecen comunicados mediante un mensaje del tipo WINDOW UPDATE (ver detalles en el Cuadro 2.6). Los pares receptores de este mensaje le deben contestar al primero con un mensaje BITFIELD, pues es posible que hayan novedades para satisfacer las necesidades de esta nueva ventana. La comunicación entre pares es dinámica. Si durante dos minutos seguidos dos pares no intercambian mensaje alguno, ambos deben enviarse un mensaje de tipo KEEPALIVE (ver la definición de este mensaje en el Cuadro 2.7). En caso contrario, la conexión se asumirá cerrada.

Cuadro 2.4: Definición del mensaje de “BITFIELD”. Este mensaje informa sobre las piezas que dispone un par específico en su ventana deslizante. Tiene largo variable según el tamaño de la ventana. Se envía inmediatamente luego del proceso de HANDSHAKE o como respuesta al mensaje WINDOW UPDATE. Las fuentes (seeders) envían un Bitfield nulo ($len=0$) en estos mensajes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Mensajee largo	$5 + X$
<i>id</i>	1	Identificador de mensaje	5
<i>offset</i>	4	Base del bitfield	N/A
<i>bitfield</i>	X	Secuencia de bits, donde cada bit anuncia si el par tiene o no la pieza. Cada pieza es identificada con respecto al offset (el N -ésimo bit representa la pieza con $ID = offset + N$)	N/A

Cuadro 2.5: Definición del mensaje “HAVE”. Este mensaje reconoce la recepción de un nuevo chunk. Su largo es fijo, de 10 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	9
<i>id</i>	1	Identificador del mensaje	4
<i>piece_id</i>	4	Identificador del chunk recibido	N/A
<i>ABI</i>	4	Value del ABI actual del par	N/A

Cuadro 2.6: Definición del mensaje “WINDOWUPDATE”. Este mensaje es enviado cada vez que un par cambia su ventana. Tiene un tamaño fijo de 6 bytes. Debe responderse con un mensaje BITFIELD.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	5
<i>id</i>	1	Identificador del mensaje	11
<i>offset</i>	4	Nueva base del bitfield	N/A

Cuadro 2.7: Definición del mensaje “KEEPALIVE”. Utilizado para mantener una conexión activa. Su largo es de 1 byte.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	0

2.3.8.2. Mensajes de Intercambio de Piezas

Ahora nos concentraremos en el mecanismo con el que se intercambian las piezas. Hasta ahora hemos visto cómo los pares intercambian información de sus estados (qué piezas tiene cada par). Con esta información los pares pueden decidir el interés de descargar una pieza de un par. Llegado el caso, el par debe enviar un mensaje de tipo INTERESTED (ver detalles en Cuadro 2.8). Habiendo enviado este mensaje, si ya no se está más interesado (posiblemente por recibir esta misma pieza de otra fuente, o haber salido de la ventana por exceso de tiempo), el mensaje correspondiente para indicar tal situación es NOT INTERESTED (Cuadro 2.9).

Cuadro 2.8: Definición del mensaje “INTERESTED”. Este mensaje es utilizado para mostrar el interés de descargar datos de otro par. Su largo es fijo e igual a 2 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	1
<i>id</i>	1	Identificador del mensaje	2

Cuadro 2.9: Definición del mensaje “NOT INTERESTED”. Este mensaje se utiliza por los pares para establecer desinterés en descargar de algún otro par. Su largo es fijo e igual a 2 bytes. Se envía solamente cuando un mensaje INTERESTED ha sido previamente enviado (del mismo emisor al mismo receptor).

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	1
<i>id</i>	1	Identificador del mensaje	3

El protocolo especifica que para que el par *A* pueda descargar una pieza del par *B*, es necesario que *A* esté interesado en descargar de *B*, y a su vez que este último se lo permita. Los pares rechazan o bien aceptan la solicitud de descarga mediante los mensajes CHOKE y UNCHOKE respectivamente (ver la definición de los mensajes en los Cuadros 2.10 y 2.11 respectivamente).

Cuadro 2.10: Definición del mensaje “CHOKE”. Si el par *A* envía este mensaje al par *B*, el último ya no puede descargar desde el primero. Tiene un largo fijo de 2 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	1
<i>id</i>	1	Identificador del mensaje	0

Cuadro 2.11: Definición del mensaje “UNCHOKE”. Si el par *A* envía este mensaje al par *B*, se permite que el último descargue del primero. Tiene un largo fijo de 2 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	1
<i>id</i>	1	Identificador del mensaje	1

Cuando un par es autorizado por otro para descargar (porque el primero envió un mensaje INTERESTED y el segundo un UNCHOKE), el primero puede solicitar partes de una pieza (denominados segmentos o “slices”, al igual que en BitTorrent) por medio de un mensaje REQUEST (por más detalles ver Cuadro 2.12).

Cuadro 2.12: Definición del mensaje “REQUEST”. Este mensaje se utiliza por los pares para solicitar la descarga de un fragmento de una pieza. Su largo es fijo, igual a 14 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	13
<i>id</i>	1	Identificador del mensaje	6
<i>piece_id</i>	4	Identificador de la pieza solicitada	N/A
<i>begin</i>	4	Inicio del fragmento (en bytes con respecto del largo de pieza)	N/A
<i>length</i>	4	Largo del fragmento (en bytes)	N/A

Nótese que si el par que está descargando una pieza es más rápido que su semilla (un superpar) puede pasar que éste pida una pieza que no haya sido generada aún. En esta situación

el transmisor puede enviar un mensaje DON'T HAVE (Cuadro 2.13).

Cuadro 2.13: Definición del mensaje “DON'T HAVE”. Este mensaje se utiliza por las fuentes (seeders) para responder solicitudes sobre piezas no generadas aún. Su largo es fijo, igual a 6 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	5
<i>id</i>	1	Identificador del mensaje	10
<i>piece_id</i>	4	Identificador de la pieza	N/A

Un par siempre puede cancelar la solicitud de un slice mediante el mensaje CANCEL. Finalmente, un par que recibe el mensaje REQUEST envía el slice utilizando el mensaje PIECE (las definiciones de los mensajes CANCEL y PIECE se encuentran en los Cuadros 2.14 y 2.15 respectivamente).

Cuadro 2.14: Definición del mensaje “CANCEL”. Con este mensaje los pares cancelan la solicitud de un slice de una pieza. Su largo es fijo e igual a 14 bytes.

Campo	Tamaño (bytes)	Descripción	Valor
<i>len</i>	1	Largo del mensaje	13
<i>id</i>	1	Identificador del mensaje	8
<i>piece_id</i>	4	Identificador de la pieza cancelada	N/A
<i>begin</i>	4	Comienzo del slice (en bytes con respecto al largo de pieza)	N/A
<i>length</i>	4	Largo del slice (en bytes)	N/A

Cuadro 2.15: Definición del mensaje “PIECE”. Con este mensaje se envía el slice de una pieza. Posee largo variable (indicado por el campo len). Es la respuesta al mensaje REQUEST.

Field	Size (bytes)	Description	Value
<i>len</i>	1	Largo del mensaje	$9 + X$
<i>id</i>	1	Identificador del mensaje	7
<i>piece_id</i>	4	Identificador de pieza	N/A
<i>begin</i>	4	Comienzo de un slice (en bytes con respecto al largo de la pieza)	N/A
<i>block</i>	X	Dato del slice	N/A

2.3.8.3. Comunicación exitosa entre pares

En resumen, una comunicación exitosa entre pares consiste en los siguientes eventos ordenados temporalmente. Una vez que el par *A* recibe el ABI y la lista del tracker, difunde el mensaje HANDSHAKE con los pares de su lista para identificar sus pares vecinos activos. Consideremos otro par de la lista que responde con HANDSHAKE a *B*. A continuación cada uno intercambia un mensaje BITFIELD, que envía un vector de campos de bits que indican las piezas que tiene para ofrecer dentro de la ventana del otro par. Supongamos que existe alguna pieza que *B* posee y *A* no; luego *A* hace llegar a *B* un mensaje INTERESTED. En su posibilidad de otorgar piezas a *A*, el par *B* lo desbloquea mediante un mensaje UNCHOKE. El par *A* está ahora en condiciones de solicitar una pieza que le falta, mediante el mensaje REQUEST. Entonces, *B* otorga en slices esta pieza, con el mensaje PIECE. A medida que *A* recibe slices exitosamente, debe indicar la actualización de su ventana mediante WINDOWUPDATE. El diagrama que se muestra en la Figura 2.3 ilustra el proceso de comunicación exitosa. En cada mensaje intercambiado se señalan algunos parámetros relevantes. Por más detalles de todos los parámetros de cada mensaje se remite al lector a los cuadros correspondientes del apartado anterior.

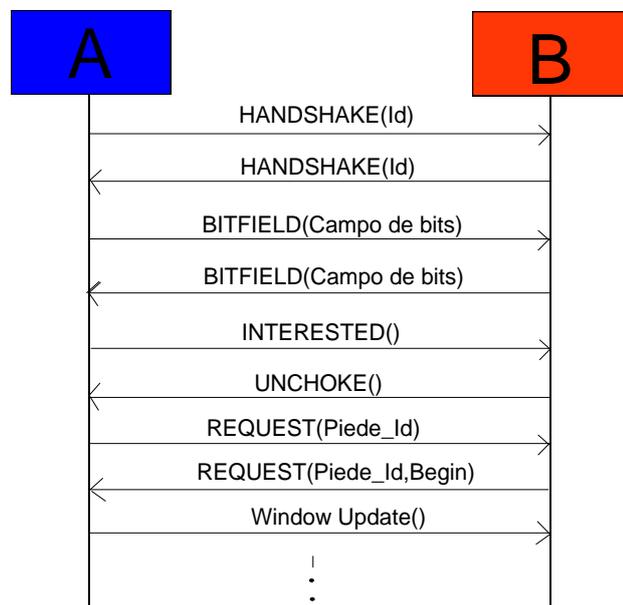


Figura 2.3: Comunicación exitosa entre pares. La secuencia continúa con mensajes WINDOWUPDATE y PIECE intercalados. Nótese que cada vez que *A* obtiene una pieza, avisa mediante un HAVE a los restantes pares que puedan beneficiarse con la misma (*B* no se incluye en este grupo). A su vez *A* mantiene la comunicación con otros pares y el tracker.

2.3.9. Estrategias aplicadas al Protocolo

Describiremos aquí dos estrategias que se pueden aplicar en el protocolo. La primera concierne a la selección de pares para realizar el intercambio de piezas, y la segunda la selección de piezas para descargar.

2.3.9.1. Estrategia de Selección de pares

La estrategia para decidir qué pares van a estar aptos para descargar datos es la misma que aquella utilizada en BitTorrent: “tit-for-tat”, que como hemos visto anteriormente, premia con piezas a los pares más generosos. A esto se le agrega la política “optimista” para explorar la red e integrar nuevos pares, ya introducida en BitTorrent bajo el nombre de “optimistic unchoking”. Se selecciona entonces un par al azar, sin tomar en cuenta cuan generoso fue en el pasado. Más precisamente, el tiempo se divide en ranuras numeradas de T segundos (comúnmente 10 segundos) cada una. Al final de cada ciclo i se habilitan a 4 pares para descargar. Si iT es múltiplo de 3, se aplica tit-for-tat para seleccionar a los 3 mejores pares y el otro par se elige aplicando optimistic unchoking. En caso contrario, los 4 pares se eligen mediante tit-for-tat.

2.3.9.2. Estrategia de selección de piezas

Consiste en decidir qué piezas serán solicitadas: una vez que necesitamos una nueva pieza, cuál pedir a continuación. En BitTorrent vimos que se aplica la estrategia de pedir la “primera más rara” (Rarest First), procurando la uniformidad en la disponibilidad de piezas. En el caso presente de streaming de video en tiempo real, se ha observado que esta estrategia repercute en altos tiempos de latencia inicial (tiempo de buffering), lo que no es un comportamiento deseado. Aquí es donde se debe hallar un compromiso entre la latencia inicial y la continuidad en la reproducción. En el Capítulo 5 se muestra que la estrategia golosa (Greedy, que consiste en solicitar las piezas más próximas a la línea de reproducción) no logra la escalabilidad en GoalBit. Es por esta razón que en GoalBit se aplica una aproximación híbrida entre ambas técnicas. En la Figura 2.4 se puede ver que se definen 3 rangos del buffer con respecto al índice de ejecución, denominados “urgente”, “usual” y “futuro”. Entonces, si una pieza faltante se halla en el sector urgente del buffer, esa pieza es solicitada (si hay más de una pieza faltante se solicita la más próxima a la línea de reproducción). En caso contrario, si todas las piezas urgentes están descargadas, la selección de pieza se realiza en base al resultado de un muestreo de una variable aleatoria exponencial que da mayor peso al rango usual, pero permitiendo solicitar chunks en el rango futuro, de vez en cuando.

Dado que el sector urgente en redes de streaming se percibe como una necesidad, lo más que se puede hacer en este es variar su largo. Por otra parte, se observa que la curva exponencial sigue manteniendo prioridad a las piezas cercanas a la línea de reproducción en el sector usual, por lo que la técnica se muestra como semigolosa.

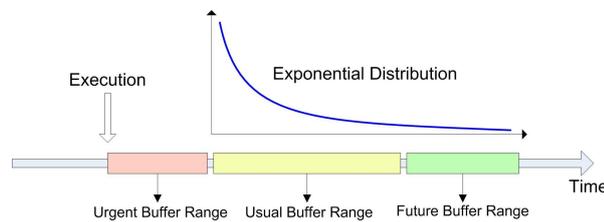


Figura 2.4: Estrategia de selección de piezas.

2.4. Clásicas Estrategias de Selección de Piezas

Existen dos elementos esenciales en la definición de un protocolo de cooperación entre pares. El primero es la política con la cual se eligen los pares con los cuales establecer comunicación, conocida como estrategia de selección de pares. La segunda es la estrategia de selección de piezas a descargar.

Hay dos clásicas estrategias de selección de piezas, denominadas Rarest First (primer más raro) y Greedy (golosa). La primera procura maximizar la información neta de la red, consultando por las piezas más raras. Esto tiene la finalidad de buscar homogeneidad de la disponibilidad de las piezas diferentes en la red, procurando así la escalabilidad. Existe un concepto goloso bien diferenciado respecto a la estrategia que consulta por el primero más raro. Desde el punto de vista del usuario, el interés radica en disponer en todo momento de la pieza que se va a reproducir a continuación. Entonces, la estrategia “Greedy” o golosa consiste en solicitar las piezas que se hallan más próximas a la línea de reproducción. Probaremos matemáticamente que la estrategia Rarest First posee latencias inaceptables para la distribución de video en vivo debido a sus altas latencias. Por otra parte, Greedy posee baja continuidad de reproducción.

Esta tesis profundiza el diseño de nuevas estrategias de selección de piezas. La motivación esencial radica en las debilidades que se le atribuyen a las clásicas estrategias, que hacen inaplicables las mismas a redes de despliegue de video en vivo. En el Capítulo 3 veremos un modelo matemático cuyo principal potencial es la simplicidad para comprender la cooperación entre pares. Este modelo a su vez propone una manera de medir la calidad de una estrategia de selección de piezas, basada en los parámetros de mayor impacto en la calidad del video streaming, que son la continuidad de reproducción y el tiempo de buffering, dejando abierta la búsqueda de una estrategia de selección de piezas óptima. Se dejan en evidencia en este capítulo las carencias de las estrategias de selección de piezas Greedy y Rarest First, hechos que promueven el diseño de nuevas estrategias. El Capítulo 4 se centra entonces en hallar estrategias de alta calidad, y es el tema central de esta tesis. Por último, en el Capítulo 5 se busca interpretar estos resultados teóricos para proponer estrategias sobre la plataforma de GoalBit. Por los motivos aquí expuestos, buscaremos aplicar las estrategias de selección de piezas obtenidas en el Capítulo 4 en los sectores usual y futuro de GoalBit, permitiendo posiblemente variar el largo del sector urgente, para analizar si existen mejoras. También se discuten posibles correlaciones con la estrategia de selección de pares y presencia de parásitos en la red, elemento que no se considera en el modelo estudiado en los Capítulos 3 y 4.

2.5. Conclusiones

Hemos revisado el protocolo BitTorrent, enumerando las características más destacadas en cuanto a técnicas de cooperación refiere. Este protocolo es bien apto para la descarga de archivos, logrando escalabilidad debida en gran parte a la estrategia Rarest First y al manejo de incentivos conjugando tit-for-tat y optimistic unchoking. Si bien estos conceptos se pueden traducir directamente a redes de streaming, en estas últimas la estrategia Rarest First es deficiente, pues la entrega de piezas en tiempo real requiere dar prioridad a piezas más próximas a la línea de reproducción.

GoalBit mantiene la filosofía de BitTorrent, ahora cambiando tal como es necesario el mecanismo de solicitud de piezas, brindando prioridad a las más cercanas al play. Define tres sectores de prioridades: urgente, usual y futuro. Se hace uso de un muestreo exponencial para determinar la siguiente pieza a solicitar en los sectores usual y futuro, solamente cuando se dispone de todas las piezas urgentes. Intuitivamente, es un plan razonable, pues se debe dar prioridad a las piezas a ser reproducidas inmediatamente, pero no se debe caer en Greedy por su no escalabilidad. Es cuestionable aquí la optimalidad de esta técnica.

En el Capítulo 3 veremos un modelo matemático que se centra en la cooperación y los distintos mecanismos de selección de piezas. Ya en el Capítulo 4 se proponen nuevas estrategias de selección de piezas, como también distintas medidas de desempeño, siempre teniendo presente los parámetros de mayor impacto en el streaming de video. El Capítulo 4 propone nuevas estrategias de selección de piezas con alta calidad sobre un modelo teórico. El dinamismo y la heterogeneidad de los pares de estas redes no se contemplan en este modelo. Se analiza entonces en el Capítulo 5 su desempeño sobre la red real GoalBit, sin alterar la prioridad de solicitud de piezas en el sector urgente. Sí desplazando el comportamiento probabilístico presente hasta el momento en los sectores 2 y 3, con nuevas estrategias de piezas determinísticas.

Parte III

MODELO DE COOPERACION EN REDES P2P

Capítulo 3

Modelo Matemático de Cooperación

3.1. Introducción

En este capítulo estudiaremos más en detalle la cooperación de pares a nivel de buffer. Se explica el modelo matemático introducido por primera vez en [97]. Este modelo tiene la desventaja de no capturar la heterogeneidad y diversidad del número de pares presentes en la red. No obstante, al ser simple permite analizar más detalladamente mecanismos de intercambio de piezas de video y cooperación. En [17] presentamos una extensión del análisis matemático brindado en [97], habiendo introducido una nueva familia de estrategias de selección de piezas, y un procedimiento sistemático de búsqueda dentro de ellas, definiendo matemáticamente un concepto de optimalidad. La profundización en técnicas de cooperación de piezas en base a este modelo matemático se manifiesta en [18]. Aquí se propone un nuevo problema de optimización combinatoria que captura la calidad de experiencia del streaming de video, y un Algoritmo de resolución basado en optimización por colonia de hormigas y búsqueda local.

Este capítulo contiene básicamente el modelo matemático a estudiar, estrategias de selección de piezas definidas con anterioridad y críticas fundadas que evidencian sus debilidades, hecho que promueve a interesarse en el diseño de nuevas estrategias de selección de piezas.

En la Sección 3.2 se muestra el modelo matemático tal como se puede encontrar en [97]. Este modelo deja abierto el problema de diseño de una estrategia de selección de piezas con altas continuidades de reproducción y bajas latencias.

La Sección 3.3 analiza dos estrategias clásicas, denominadas Rarest First y Greedy, y una estrategia mixta diseñada como combinación de éstas. Se introduce además diferentes maneras de modelarlas analíticamente. Las tres son propuestas del artículo original [97].

La Sección 3.4 contiene un análisis matemático que justifica las causas del bajo desempeño de las anteriores estrategias de selección. Se podrá comprender así la pobre continuidad que produce la estrategia golosa o Greedy, como también las altas latencias que presenta la aplicación de Rarest First. Una estrategia híbrida que considera las bondades de cada una se

denomina aquí Mixta. Si bien esta consigue un compromiso entre latencia y continuidad deja margen de mejoras.

En la Sección 3.5 definimos una nueva estrategia de selección de piezas, que denominamos Golosa Ponderada. Si bien esta no logra el desempeño deseado (siendo incluso inferior al de las estrategias antes mencionadas) ilustra un primer diseño básico en la búsqueda de estrategias de selección globalmente eficientes. La principal contribución de esta tesis se encuentra en el Capítulo 4. En el mismo se define una nueva familia de estrategias, como también algoritmos de obtención automática de miembros de alta calidad.

Por último, la Sección 3.6 contiene las conclusiones de este capítulo.

3.2. Modelo Matemático

Consideremos una red en la que un servidor posee el contenido de video original, y M pares que desean obtener y reproducir el mismo. A los efectos de utilizar eficientemente el ancho de banda y la cooperación de los pares, el servidor corta el video en piezas, y en cada ranura de tiempo elige un par X al azar dentro de la red, para asignarle esa pieza. Solamente el par X va a ser el beneficiado de obtener esa pieza sin necesidad de realizar consultas a sus otros pares. Todos los pares poseen igual capacidad de almacenamiento de piezas, denotado con N . Se supone que todos los nodos de la red intercambian piezas con la misma estrategia de cooperación. A su vez, un par puede completar una consulta y obtener una pieza (o no), en un tiempo inferior al time slot, que se corresponde con el tiempo de reproducción de una pieza. Todos los pares están sincronizados, es decir, despliegan la pieza más antigua (en la posición N , ver Figura 3.1) dentro del buffer (si es que la disponen), al mismo tiempo. El usuario percibe un defecto o error en la imagen cada vez que no se disponga de la pieza a ser reproducida.

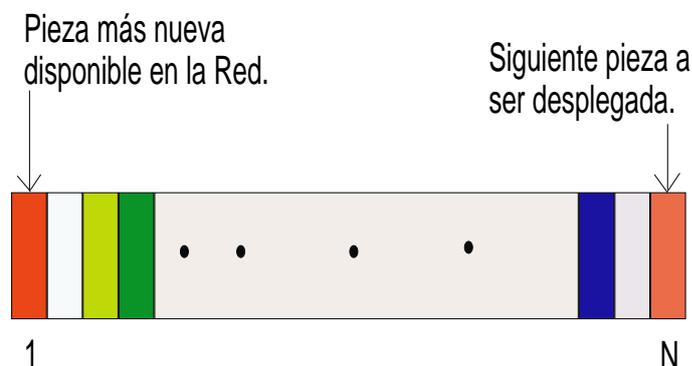


Figura 3.1: Modelo de buffer en cada par. La posición 1 representa la pieza de video más nueva en la red y la posición N la siguiente pieza a ser desplegada. Obsérvese que todos los pares están sincronizados en el consumo de buffer.

El intercambio de piezas funciona de la siguiente manera : el par h escoge al azar a otro par k perteneciente a la red, y solicita una pieza que no dispone. Si k tampoco posee esta pieza, h puede consultar por otra. El proceso se repite hasta que h consigue una pieza y la consulta resulta exitosa, o bien no consigue pieza, refiriéndonos en este último caso a una consulta fracasada. La consulta finaliza y el par fracasado h no puede contactarse con otro nuevo par durante el mismo time slot.

Llamemos p_i a la probabilidad de que un par tenga la pieza correcta en el i -ésimo espacio del buffer, $B(i)$. Asumiendo estado estacionario es que por simetría, el vector de p_i 's es idéntico para cada par de la red. Nos es de interés ahora estudiar en qué medida la cooperación logra mayores probabilidades de ocupación. El buffer $B(j)$ va a tener pieza en la ranura de tiempo t_i en dos situaciones. La primera es cuando el buffer $B(j - 1)$ tenía pieza en el time slot anterior t_{i-1} , ya que las piezas aumentan un índice de buffer en cada ranura de tiempo. La segunda es cuando durante la consulta transcurrida en la ranura de tiempo t_{i-1} se accede a esta pieza. En consecuencia, en este proceso de cooperación un par puede ser beneficiado por una pieza ya sea directamente por el servidor, o bien por la adquisición de la pieza en una consulta. Si denotamos con u_i a la probabilidad de adquisición de la pieza en la posición de buffer i mediante una consulta, tenemos que:

$$p_1 = \frac{1}{M}$$

$$p_{i+1} = p_i + u_i, \forall i = 1 \dots N - 1.$$

Esta ecuación expresa que el buffer $B(i + 1)$ va a estar lleno en la ranura de tiempo t_{j+1} si en el tiempo anterior t_j se disponía de la pieza i o bien si se ha logrado obtener esta pieza en la consulta transcurrida en el tiempo t_j . A continuación plantearemos una expresión analítica simple que nos permitirá hallar el vector de probabilidades de ocupación p_i para cada estrategia de selección. Llamemos h al par solicitante y k al solicitado. Consideremos los siguientes 3 eventos:

- (1) $Q(h, i)$: el par h quiere la pieza i , pues su buffer $B(i)$ está vacío.
- (2) $T(k, i)$: el par k tiene la pieza i .
- (3) $S(k, h, i)$: el par k dispone de la pieza i , y usando una estrategia de selección, no hay otra pieza más deseable por el par h .

A partir de estos eventos es posible expresar el término de cooperación de la siguiente manera:

$$u_i = P(Q(h, i) \cap T(k, i) \cap S(k, h, i)) = P(Q(h, i))P(T(k, i)/Q(h, i))P(S(k, h, i)/Q(h, i) \cap T(k, i)) \quad (3.1)$$

Atendamos ahora cada uno de los factores:

- La probabilidad de que el par h desee el buffer en la posición i es sencillamente la probabilidad de que no tenga esa pieza: $P(Q(h, i)) = 1 - p_i$.

- Vamos a asumir que la probabilidad de que el par k disponga de la pieza i es independiente de que el par h quiera esa pieza. Esto implica que: $P(T(k, i)/Q(h, i)) \approx P(T(k, i)) = p_i$.

Obsérvese que esta simplificación es más fidedigna cuanto mayor cantidad de pares hay presentes en la red. Puesto que la redes P2P frecuentemente contienen a centenas o miles de pares, esta aproximación es apropiada en general.

- Las piezas se distribuyen independientemente en la red. Asumimos entonces que la distribución de probabilidades de la pieza en la posición i no se ve fuertemente influenciada por el conocimiento del estado de otras posiciones. Entonces:

$$s_i = P(S(k, h, i)/Q(h, i) \cap T(k, i)) \approx P(S(k, h, i)) \quad (3.2)$$

A partir de las consideraciones anteriores, sustituyendo en (3.1) se obtiene la importante expresión, que vincula la probabilidad de ocupación con la función de estrategia de selección:

$$p_1 = 1/M; \quad (3.3)$$

$$p_{i+1} = p_i + (1 - p_i)p_i s_i, \quad i = 1, \dots, N - 1 \quad (3.4)$$

Este es el modelo matemático que se estudia en la extensión de este capítulo y el siguiente. Antes de dar definiciones relativas a la calidad de las distintas estrategias de selección de piezas es importante comprender la anterior recursión, que define el vector de probabilidades de ocupación p_i dado un mecanismo de selección de piezas establecido a priori. La probabilidad de que un par dado sea elegido por el servidor es p_1 . Puesto que hay M pares en la red y el servidor elige al azar a los pares, $p_1 = 1/M$. Además todas las piezas avanzan un índice en el buffer en cada ranura de tiempo. Entonces, la posición $i + 1$ se va a ocupar en el tiempo t_{i+1} de dos maneras distintas. La primera ocurre cuando en la ranura de tiempo anterior t_i se tenía anteriormente ocupado al buffer en posición i (vía promoción de la pieza con el tiempo). La segunda ocurre cuando se cumplen tres condiciones: el par solicitante no tiene la pieza (con probabilidad $1 - p_i$), el solicitado sí la tiene (con probabilidad p_i) y además el par elige la pieza i para su descarga, cuya probabilidad es s_i . Reuniendo las anteriores observaciones, y recordando que en estado estacionario la probabilidad de ocupación no depende del tiempo, se obtiene la recursión 3.4. La función s_i se denomina *estrategia de selección de piezas*, pues está sujeta al diseño del mecanismo de consulta, en el que se debe establecer el orden de solicitud de las piezas a priori (antes de que el servidor inicie la distribución).

El Capítulo 4 se enfoca en los mecanismos de diseño que logren altas continuidades de reproducción de video, manteniendo a la vez bajas latencias iniciales. Estos dos conceptos se definen formalmente a continuación.

Definición 3.2.1 *La continuidad de reproducción es medida mediante $C = p_N$, y es la probabilidad de tener la pieza inmediata a ser reproducida.*

Iniciando con un buffer vacío y considerando que los pares obtienen a lo sumo una pieza por consulta, se esperarán $\sum_{i=1}^N p_i$ ranuras de tiempo para obtener el número de piezas del estado

estacionario. Si se comienza a reproducir antes de este tiempo cabe esperar mayor cantidad de discontinuidades en la reproducción al inicio. Además, no hay razones para comenzar la reproducción más tarde.

Definición 3.2.2 *La latencia de inicio del video se mide con:*

$$L = \sum_{i=1}^N p_i.$$

Y representa el tiempo (medido en slots) que se requiere para alcanzar el estado estacionario, partiendo de un buffer vacío.

Este tiempo es usualmente configurado como el tiempo de buffering, y naturalmente, se desea reducir. Se observa que la latencia coincide con el número medio de piezas que dispone cada par en estado estacionario. Esto es así dado que un par que inicia con un buffer vacío debe esperar en media tanto tiempo como número medio de piezas se disponga, dado que toda pieza le sirve.

3.3. Estrategias Clásicas y una Hibridación

Esta sección presenta dos clásicas estrategias de selección de piezas, denominadas Rarest First y Greedy, y una nueva estrategia híbrida, atendiendo a sus desempeños. Rarest First goza de una prestigiosa popularidad actualmente en sistemas de descarga, siendo utilizada desde sus orígenes por BitTorrent [54]. Como su nombre indica, básicamente consiste en seleccionar una pieza faltante que el par contactado tenga, y buscando inicialmente piezas lejos de la línea de reproducción (estas piezas son las más raras, debido a la monotonía de p):

$$s_i = \left(1 - \frac{1}{M}\right) \prod_{j=1}^{i-1} (p_j + (1 - p_j)^2) \quad (3.5)$$

Esta expresión tiene una clara interpretación: en todas las posiciones de buffer j anteriores a i , el par solicitante ya tiene la pieza j (con probabilidad p_j) o bien no la tiene pero el par solicitado tampoco (cuya probabilidad es $(1 - p_j)^2$). Vale apreciar que el par solicitante obtendrá la pieza en la posición i si además el solicitado tiene la pieza i y el solicitante tampoco, tal como lo expresa el término de cooperación $(1 - p_i)p_i s_i$ del modelo matemático 3.4.

Por otra parte, es bien sabido que Greedy logra bajas latencias, pero no es escalable como Rarest First [89, 97]. La estrategia Greedy es idéntica a la anterior, pero consulta primero por las piezas más cercanas a la reproducción (ver Figura 3.3):

$$s_i = \left(1 - \frac{1}{M}\right) \prod_{j=i+1}^{N-1} (p_j + (1 - p_j)^2) \quad (3.6)$$

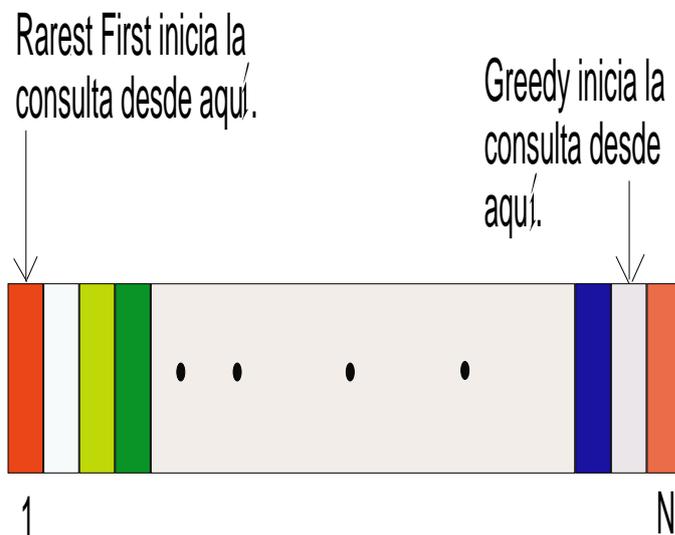


Figura 3.2: Orden de consulta en las estrategias Greedy y Rarest First. Rarest First comienza la consulta en el índice 1 y avanza hacia la dirección de reproducción, incremental en índices. Por su parte Greedy comienza en el índice $N - 1$ y continúa la consulta alejándose de la línea de reproducción

La Figura 3.3 muestra gráficamente la relación tamaño de buffer vs. probabilidad de ocupación para las estrategias Rarest First y Greedy en el caso de 1000 pares y un buffer de dimensión 40.

Una mezcla de estas dos últimas estrategias es posible, cortando el buffer en un índice dado m : $1 \leq m \leq N$ y aplicando la estrategia de Rarest First en la primera partición, y Greedy en la otra. Esta estrategia ofrece una latencia más reducida que Rarest First y buena continuidad. Se sugiere revisar [97] por más detalles.

3.4. Análisis de Desempeño

Las expresiones 3.5 y 3.6 se pueden simplificar notoriamente mediante inducciones:

Lema 3.4.1 En Rarest First vale que $s_i = 1 - p_i, \forall i = 1, \dots, N - 1$

Prueba. Por definición, $s_1 = 1 - 1/M = 1 - p_1$, por lo que se cumple el paso base. Supongamos que $s_j = 1 - p_j$ para algún $j \in \{1, \dots, N - 2\}$. Luego, a partir de la expresión 3.5 para

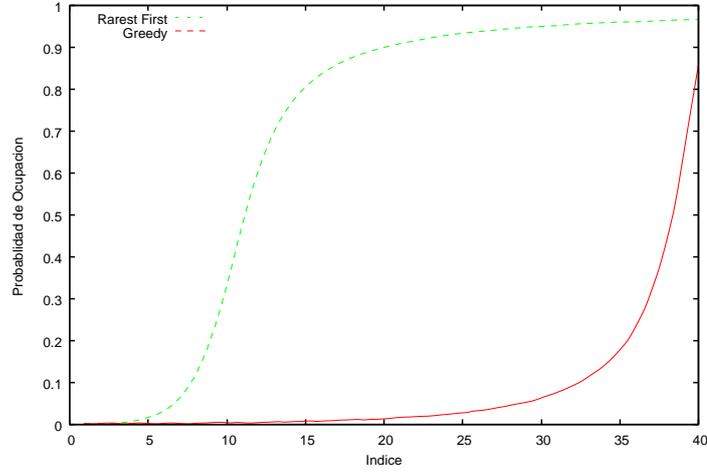


Figura 3.3: Probabilidades de ocupación de las estrategias clásicas para 1000 pares y tamaño de buffer 40.

Rarest First se obtiene que:

$$s_{j+1} = s_j[p_j + (1 - p_j)^2] = s_j(1 - p_j(1 - p_j)) = s_j - (p_{j+1} - p_j) = 1 - p_{j+1} \quad (3.7)$$

QED

Análogamente, en Greedy se puede probar que:

$$s_i = 1 - p_N + p_{i+1} - p_1, \forall i = 1, \dots, N - 1 \quad (3.8)$$

3.4.1. Performance de Rarest First

Sustituyendo la expresión simplificada para Rarest First en (3.4) se tiene que:

$$p_1 = 1/M; \quad (3.9)$$

$$p_{i+1} = p_i + (1 - p_i)^2 p_i, \quad i = 1, \dots, N - 1 \quad (3.10)$$

La expresión (3.10) permite analizar la continuidad y latencia mediante una recursión de orden 1. Obsérvese que es posible obtener el siguiente índice de la probabilidad de ocupación dado el actual, evaluando un polinomio. Si bien aún no se ha encontrado una expresión explícita en n para p_n , daremos a continuación unos resultados analíticos sencillos que explican la obtención de buena continuidad de Rarest First, como también su pobre desempeño en cuanto a latencia refiere.

Lema 3.4.2 *En Rarest First, valen las tres afirmaciones siguientes:*

- i) $p_i < 1, \forall i \leq N$ (continuidad imperfecta).
- ii) $p_i < p_{i+1}, \forall i < N$ (monotonía estrictamente creciente).
- iii) $\lim_{N \rightarrow \infty} p_N = 1$ (tendencia perfecta a mayor capacidad).

Prueba.

- i) Sabemos que $p_1 = \frac{1}{M} < 1$, pues estamos considerando una red de cooperación, con varios pares, ($M > 1$). Si admitimos ahora que $p_h < 1$ para cierto $h : 0 < h < N$, usando (3.10) y el hecho que $p_h > 0$ tenemos que:

$$p_{h+1} = p_h + (1 - p_h)^2 p_h < p_h + (1 - p_h) = 1, \quad (3.11)$$

- ii) Por (3.10) y (i): $p_{i+1} - p_i = (1 - p_i)^2 p_i > 0, \forall i < N$.
- iii) Por (ii) la sucesión $\{p_N\}$ es monótona creciente, y por (i) está acotada por 1. Por lo tanto, su límite existe. Si denotamos con $\alpha = \lim_{N \rightarrow \infty} p_N$ y tomamos límite en ambos miembros de (3.10), entonces:

$$\alpha = \alpha + (1 - \alpha)^2 \alpha \quad (3.12)$$

Despejando se ve que $\alpha = 0$ o bien $\alpha = 1$. Pero α no es cero puesto que $p_1 = \frac{1}{M}$ y $\{p_N\}$ es monótona creciente.

QED

Teorema 3.4.3 *El orden de convergencia de Rarest First es lineal, y su tasa de convergencia es 1.*

Prueba. Por el Lema 3.4.2 sabemos que Rarest First converge a 1. Definamos el error en el paso n mediante $e_n = 1 - p_n$. Luego, a partir de la (3.10):

$$h_n = \frac{e_{n+1}}{e_n} = \frac{1 - p_n - (p_{n+1} - p_n)}{1 - p_n} = 1 - p_n(1 - p_n) \quad (3.13)$$

h_n está bien definido, pues por el Lema 3.4.2 se ve que $e_n > 0, \forall n > 0$. El resultado se prueba tomando límites en ambos miembros, y usando que $\lim_{N \rightarrow \infty} p_N = 1$, tal como expresa el Lema 3.4.2.

QED

Además, como $0 < p_n < 1, \forall n > 0$ se observa que $\frac{3}{4} \leq h_n < 1, \forall n > 0$. Para tamaños limitados de buffer, definamos $r = \max_{n: 1 \leq n < N} \{h_n\} < 1$. Entonces tenemos que $e_N < r^{N-1} e_1 = r^{N-1} (1 - \frac{1}{M})$. Esto explica que el error global en Rarest First se reduce a tasa exponencial, y la anterior potencia es tan sólo una cota superior de la discontinuidad $e_N = 1 - p_N$.

Existe un mínimo absoluto de la función $h_n = 1 - p_n(1 - p_n)$ en el índice cuya probabilidad de ocupación es más próxima a $\frac{1}{2}$. Es en el paso entre este índice y el siguiente donde se logra la mayor reducción del error global e_n , y el respectivo factor de reducción es cercano (o igual, solamente cuando existe $i : p_i = \frac{1}{2}$) a $\frac{3}{4}$.

Las propiedades anteriores muestran el potencial de Rarest First para lograr altos valores de continuidad. No obstante, esta estrategia también se caracteriza por altos valores de latencia, como se explica a continuación.

Consideremos el polinomio que expresa la magnitud del salto entre dos índices consecutivos en Rarest First:

$$P(x) = x(1 - x)^2 \quad (3.14)$$

Este polinomio restringido al intervalo $[0, 1]$ presenta un máximo absoluto en $x = \frac{1}{3}$. Esto muestra que cuando la probabilidad de ocupación está en proximidades de este valor, la sucesión p_i ya aplica un salto importante, y aún resta buena cantidad de índices cuya magnitud es superior a $\frac{1}{3}$, que aportan en la latencia: $L = \sum_{i=1}^N p_i$.

Esto fundamenta los valores inaceptables de la latencia Rarest First, cuando se desea un streaming de video en tiempo real.

3.4.2. Performance de Greedy

La estrategia Greedy o golosa consulta en primera instancia por las piezas más cercanas a la línea de reproducción. Si bien esto parece natural por la mayor urgencia de ocupación con piezas de estos espacios del buffer, no mantiene su desempeño cuando el número de pares incrementa, como ocurre con Rarest First. El problema radica básicamente en el hecho de que la red crea en exceso piezas cercanas a la línea de reproducción, y no así con las piezas en las primeras posiciones del buffer. Esto genera una distribución no uniforme de la cantidad de piezas en un instante dado, y las piezas raras son muy difíciles de conseguir.

Sustituyendo la expresión (3.8) en (3.4) se tiene en Greedy que:

$$p_1 = 1/M; \quad (3.15)$$

$$p_{i+1} = p_i + (1 - p_i)p_i(1 - p_N + p_{i+1} - p_1), \quad i = 1, \dots, N - 1 \quad (3.16)$$

Al igual que Rarest First, la probabilidad de ocupación converge a 1 cuando el número de buffer tiende al infinito. La prueba es análoga. A efectos comparativos de la performance entre Greedy y Rarest First, comparemos los saltos $p_{i+1} - p_i$. Reescribiendo la expresión (3.16) y denotando p_i^G a la probabilidad de ocupación en Greedy, tenemos que:

$$p_{i+1}^G - p_i^G = (1 - p_i^G)^2 p_i^G + (1 - p_i^G) p_i^G (p_i^G + p_{i+1}^G - (p_N^G + \frac{1}{M})), \quad i = 1, \dots, N - 1 \quad (3.17)$$

Por otra parte, denotando mediante p_i^R a la probabilidad de ocupación de Rarest First, y utilizando la ecuación (3.4.1):

$$p_{i+1}^R - p_i^R = (1 - p_i^R)^2 p_i^R, i = 1, \dots, N - 1 \quad (3.18)$$

A continuación daremos un resultado que justifica la baja latencia en Greedy.

Proposición 3.4.4

$$p_i^G < p_i^{RF}, \forall i : 1 < i < i^*, \quad (3.19)$$

donde i^* es el primer índice que cumple que $p_{i^*}^G + p_{i^*+1}^G > p_N^G + \frac{1}{M}$.

Prueba. La prueba es directa por inducción sobre el conjunto $\{2, \dots, i^* - 1\}$.

Recordemos que $p_1^G = p_1^{RF} = \frac{1}{M}$. El paso base se prueba a partir de la diferencia de las expresiones (3.17) y (3.18) en $i = 1$:

$$p_2^G - p_2^{RF} = (p_2^G - p_1^G) - (p_2^{RF} - p_1^{RF}) = (1 - \frac{1}{M}) \frac{1}{M} (p_2^G - p_N^G) < 0 \quad (3.20)$$

Supongamos ahora que para cierto $h : 2 \leq h < i^* - 1$ se cumple que $p_h^G < p_h^{RF}$. Por monotónia estricta del polinomio $P(h) = h + h(1 - h)^2$ sabemos que:

$$p_{h+1}^{RF} = P(p_h^{RF}) > P(p_h^G) = p_{h+1}^G + (1 - p_h^G) p_h^G (p_h^G + p_{h+1}^G - (p_N^G + \frac{1}{M})) \quad (3.21)$$

Para completar la prueba del paso inductivo recuérdese que por definición de i^* entonces $h : p_h^G + p_{h+1}^G \leq p_N^G + \frac{1}{M}$. Sustituyendo en la expresión (3.21) resulta finalmente que:

$$p_{h+1}^{RF} > p_{h+1}^G + (1 - p_h^G) p_h^G (p_h^G + p_{h+1}^G - (p_N^G + \frac{1}{M})) \geq p_{h+1}^G \quad (3.22)$$

Esto completa la demostración del enunciado.

QED

Intuitivamente, la pobre continuidad de Greedy viene acompañada de su baja latencia: al tomar valores de probabilidad de ocupación inferiores a Rarest First hasta un índice i^* (enunciado 3.4.4), la continuidad queda limitada por

$$p_N^G = p_{i^*-1}^G + \sum_{i=i^*}^N (p_i^G - p_{i-1}^G) < p_{i^*-1}^{RF} + \sum_{i=i^*}^N (p_i^G - p_{i-1}^G)$$

3.4.3. Cotas de los Parámetros de Calidad

Antes de introducirnos en el diseño de nuevas estrategias de selección de piezas daremos dos postulados triviales, mas su aplicabilidad a cualquier estrategia a diseñar es lo que destaca su valor. El primero es pesimista, y establece que independientemente de la estrategia de selección de piezas a aplicar la continuidad de reproducción no puede alcanzar la unidad, hecho que recae en la calidad de experiencia del usuario: con frecuencia (posiblemente pequeña) va

a presenciar cortes. Esto generaliza la continuidad imperfecta enunciada bajo el Lema 3.4.2, probado únicamente para Rarest First.

El otro resultado es optimista, y afirma que la latencia inicial no puede exceder de N ranuras de tiempo independientemente de la estrategia de selección a aplicar, siendo N la misma capacidad de buffer de todos los pares. Vamos a las respectivas pruebas.

Proposición 3.4.5 *Si $M > 1$ entonces $p_N < 1$.*

Prueba. Probaremos por inducción que $p_i < 1$, $\forall i \in \{1, \dots, N\}$, cumpliéndose en particular el enunciado. El modelo matemático (3.4) implica que $p_1 = \frac{1}{M} < 1$, pues hay más de un par en la red. Esto prueba el paso base. Supongamos ahora que $p_h < 1$ para cierto $h : 1 \leq h < N$, y sea s una estrategia de selección de piezas arbitraria. Entonces el modelo matemático plantea la siguiente recursión: $p_{h+1} = p_h + (1 - p_h)p_h s_h$. Recordemos que s por definición es una probabilidad, por lo que $s_h \leq 1$. Entonces resulta que $p_{h+1} \leq p_h + (1 - p_h)p_h < p_h + (1 - p_h) = 1$, donde se utiliza finalmente la hipótesis inductiva.

QED

Una consecuencia directa de lo anterior es una cota trivial para la latencia.

Proposición 3.4.6 *Para cualquier estrategia de selección de piezas se cumple que la latencia está acotada estrictamente por N ranuras de tiempo, donde N es la capacidad de buffer de todos los pares:*

$$L = \sum_{i=1}^N p_i < N \quad (3.23)$$

La cota expresada bajo la Proposición 3.4.5 va a ser utilizada en el Capítulo 4, aplicada en particular a una familia de estrategias allí introducidas. La cota de 3.4.6 es de importancia conceptual, y se va a normalizar más adelante medidas de optimalidad que contienen la latencia en base a este resultado.

3.5. Análisis Primitivo de nuevas Estrategias

La deficiencia de las clásicas estrategias anteriormente estudiadas motiva a investigar la manera con la que se construye una estrategia de selección de piezas, a los efectos de proponer nuevas con mayor calidad.

Veremos en una primera instancia a modo ilustrativo una primera aproximación que si bien no muestra mejoras respecto a las clásicas estrategias ilustra la flexibilidad del diseño de mecanismos de selección de piezas. Luego se muestra la estrategia Mixta (híbridación de Greedy y Rarest First), que busca capturar lo mejor de las estrategias clásicas. Existen elementos que se pueden mejorar respecto de esta última estrategia mixta. Posteriormente en el Capítulo 4 se introducirá una nueva familia de estrategias y un camino para explorar mejores técnicas de selección; mostrando finalmente como resultado de esta investigación el diseño de estrategias de selección superiores en performance respecto de la estrategia Mixta, y a las relevadas en la literatura.

3.5.1. Greedy Ponderado

Veremos a efectos ilustrativos una nueva estrategia de selección que es similar a Greedy, y que le agrega a esta última una función de “deseo”, que permite continuar la consulta que parte desde el extremo de reproducción, incluso cuando el par solicitante no tiene cierta pieza y el solicitado sí la tiene. Consideremos la secuencia de deseo dada por r_i :

$$r_i = \frac{q^i}{\sum_{j=1}^{N-1} q^j}, \quad \forall i = 1, \dots, N-1 \quad (3.24)$$

Donde el parámetro $q : 0 < q < 1$, expresa el deseo por piezas más cercanas al buffer. Luego es posible definir una estrategia de selección de piezas mediante la siguiente productoria:

$$s_i = r_i \left(1 - \frac{1}{M}\right) \prod_{j=i+1}^{N-1} [p_j + (1 - p_j)^2 + (1 - p_j)p_j(1 - r_j)] \quad (3.25)$$

El orden de consulta es el mismo que el de Greedy, es decir, se revisa los espacios del buffer más cercanos a la línea de reproducción primero. Esta estrategia funciona de la siguiente manera. Los dos primeros términos tienen el mismo significado que en Greedy: el par solicitante no pide piezas en el buffer que ya tiene (probabilidad p_j), o bien aquellas que no tiene pero el par solicitado tampoco (suceso con probabilidad $(1 - p_j)^2$). Se agrega un tercer término que introduce la secuencia de deseo $((1 - p_j)p_j(1 - r_j))$, y expresa que en caso de posibilidad de admitir una nueva pieza (el par consultante no la tiene y el consultado sí) no se va a pedir esa pieza si no se desea. En otras palabras, se va a pedir la primera pieza más cercana a la línea de reproducción que sea posible y además se desee.

Se desea a continuación evaluar la calidad de la estrategia Golosa Ponderada para cada terna de entrada q , M y N . A tales efectos se puede plantear una recursión de orden uno en s planteando el cociente s_i/s_{i+1} a partir del uso de la expresión (3.25). Juntando el resultado con la ecuación (3.4) que define a las probabilidades de transición se tiene el siguiente sistema de ecuaciones para los vectores s y p :

$$p_{i+1} = p_i + (1 - p_i)p_i s_i, \quad \forall i = 1, \dots, N-1 \quad (3.26)$$

$$s_i = \frac{s_{i+1}}{q} [1 - (1 - p_i)p_i r_i], \quad \forall i = 1, \dots, N-2 \quad (3.27)$$

Este es un sistema de ecuaciones no lineal compatible determinado, que debe satisfacer las condiciones de borde $p_1 = \frac{1}{M}$ y $s_{N-1} = (1 - \frac{1}{M})r_{N-1}$ (por definición de la estrategia). Para su resolución, obsérvese que si conocemos p_N y q , estamos en condiciones de despejar p_{N-1} de la ecuación 3.26 con $i = N-1$, y mediante simple sustitución hacia atrás (despejando en la iteración i a p_{N-i} y s_{N-i} , usando alternadamente las ecuaciones (3.26) y (3.27)), las secuencias enteras s_i y p_i . El valor de continuidad p_N así tanteado va a ser el correcto cuando $p_1 \approx \frac{1}{M}$.

El desempeño de este primer diseño de estrategia de selección de piezas no es alentador. Esto tiene dos explicaciones complementarias. La primera es que dado que la secuencia r_j que mide el “deseo” es exponencial con razón q , la búsqueda se limita prácticamente a pocas piezas cercanas a la línea de reproducción, pues la estrategia de selección es proporcional a este deseo, y por lo tanto muy reducida lejos de la línea de reproducción. La segunda es que cuando un par encuentra una pieza que no posee, no hay razones para rechazarla. Es razonable que cuando encontremos una pieza faltante la pidamos automáticamente, pues en caso contrario la consulta puede fracasar (por ejemplo si esta es la única pieza que el par no posee, o si el par contactado no tiene otra pieza para ofrecer al par local).

Lema 3.5.1 *Las siguientes dos afirmaciones son válidas:*

$$a) \quad s_i < \frac{s_{N-1}}{q^{N-i-1}}, \forall i = 1, \dots, N-2$$

$$b) \quad \text{Las diferencias de probabilidades de ocupación } p_{i+1} - p_i \text{ admiten la cota superior:}$$

$$p_{i+1} - p_i < \frac{s_{N-1}}{4q^{N-i-1}}$$

Prueba. Sean M y N enteros positivos, $q \in (0, 1)$ y s la estrategia de deseo correspondiente.
a) A partir de la ecuación (3.27) obtenemos la siguiente desigualdad entre términos consecutivos de la estrategia de selección:

$$s_i = \frac{s_{i+1}}{q} [1 - (1 - p_i)p_i r_i] < \frac{s_{i+1}}{q}, \forall i = 1, \dots, N-2. \quad (3.28)$$

Una directa inducción sobre el conjunto finito $\{1, \dots, N-2\}$ muestra que:

$$s_i < \frac{s_{N-1}}{q^{N-i-1}}, \forall i = 1, \dots, N-2$$

b) Acotemos ahora s_{N-1} :

$$s_{N-1} = r_{N-1} \left(1 - \frac{1}{M}\right) < r_{N-1} = \frac{q^{N-1}}{\sum_{i=1}^{N-1} q^i} < q^{N-2} \quad (3.29)$$

Ahora aplicando (a) podemos acotar s_i :

$$s_i < \frac{s_{N-1}}{q^{N-i-1}} < q^{i-1}, \forall i = 1, \dots, N-1 \quad (3.30)$$

Esta cota en la estrategia de selección impacta directamente en los saltos de las probabilidades de ocupación:

$$p_{i+1} - p_i = (1 - p_i)p_i s_i \leq \frac{s_i}{4} < \frac{s_{N-1}}{4q^{N-i-1}}, \quad (3.31)$$

QED

El siguiente resultado permite acotar la continuidad en la reproducción ante la aplicación de la estrategia Golosa Poderada, para cualquier instancia.

Proposición 3.5.2 *En la estrategia Golosa Ponderada, la continuidad c está acotada por:*

$$c < \frac{1}{M} + \frac{(1 - \frac{1}{M})^2}{4}$$

Prueba. En base a la parte (b) del Lema 3.5.1, podemos acotar la continuidad de la reproducción:

$$c = \frac{1}{M} + \sum_{i=1}^{N-1} (p_{i+1} - p_i) < \frac{1}{M} + \frac{s_{N-1}}{4} \frac{\sum_{i=1}^{N-1} q^i}{q^{N-1}} \quad (3.32)$$

Por definición, s_{N-1} vale (recordemos que la productoria vale 1 si no alcanza factores):

$$s_{N-1} = r_{N-1} (1 - \frac{1}{M}) = (1 - \frac{1}{M}) \frac{q^{N-1}}{\sum_{i=1}^{N-1} q^i} \quad (3.33)$$

Finalmente, sustituyendo la expresión (3.33) en (3.32) se halla una inaceptable cota para la continuidad de reproducción en esta estrategia, dada por:

$$c < \frac{1}{M} + \frac{(1 - \frac{1}{M})^2}{4} \cong \frac{1}{4} \quad (3.34)$$

donde la última aproximación se toma pues el número de pares M es generalmente muy alto.

QED

Ya es evidente a partir del análisis anterior que se descarta toda posibilidad de aplicación de esta estrategia, debiéndose optar por otras alternativas. Este inicio del estudio de estrategias de selección motiva a dirigir el esfuerzo en el caso que el deseo es siempre 1, y posiblemente combinando estrategias clásicas, o bien considerando variantes del orden de consulta, como veremos en el Capítulo 4.

3.5.2. La estrategia Mixta

Los autores de [97] proponen cortar el buffer en un índice $m : 1 < m < N$, y aplicar en una primera etapa consultas según el orden de Rarest First en el rango del buffer $[1, m]$, y posteriormente el mismo orden de consultas que Greedy, con el rango faltante (m, N) . La Figura 3.5.2 contrasta el desempeño de la estrategia Mixta en relación a las clásicas para el caso de $M = 1000$ pares y tamaño de buffer $N = 40$, tomando el corte $m = 10$. Se observa que hasta el índice de buffer $i = m$, la estrategia Mixta tiene el mismo comportamiento que Rarest First. Sin embargo, luego de este índice la forma de la probabilidad de ocupación es similar a la de Greedy. Recordemos que la deficiencia principal en Rarest First proviene de acarrear continuidad, que se maximiza en el índice más próximo a $\frac{N}{3}$, que es donde ocurre la maximización del salto de índices. La estrategia Mixta se muestra como un logrado compromiso entre latencia y continuidad. La latencia se reduce considerablemente con respecto a Rarest First, manteniendo cercana su continuidad.

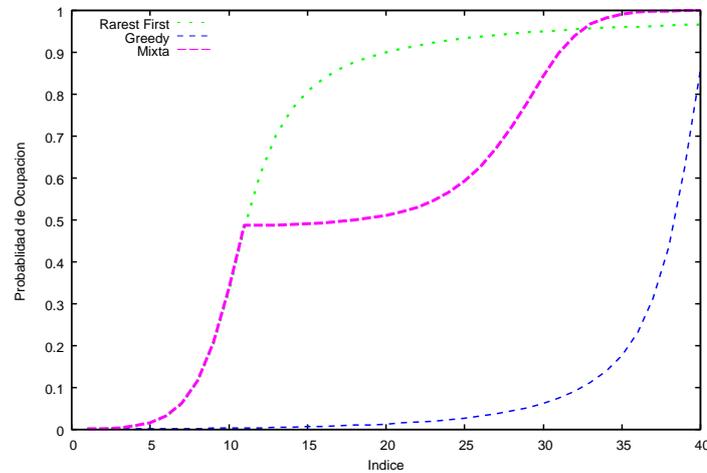


Figura 3.4: Contraste entre las estrategias Mixta, Greedy y Rarest First, para $M = 1000$, $N = 40$ y $m = 10$. La estrategia Mixta logra un interesante compromiso entre latencia y continuidad.

3.6. Conclusiones

En este capítulo se ha introducido un modelo matemático simple de cooperación entre pares. El mismo reduce el diseño de estrategias de selección de piezas a un mecanismo de orden de solicitud de piezas, establecido a priori y aplicado por todos los pares. Hemos revisado estrategias clásicas, denominadas Greedy y Rarest First. La primera consiste en realizar la consulta de las piezas iniciando por los espacios del buffer cercanos a la línea de reproducción, llamada golosa o Greedy. La segunda es la que se aplica en el protocolo BitTorrent y se llama Rarest First, solicitando las piezas más raras (más lejanas de la línea de reproducción) procurando así uniformizar la disponibilidad de piezas en la red. A su vez se definen matemáticamente dos parámetros fundamentales del streaming de video, que son la continuidad en la reproducción y latencia inicial. El análisis matemático aquí realizado deja en evidencia bajas continuidades ante la aplicación de Greedy, y altas latencias de Rarest First. Más aún, veremos en el Capítulo 5 que el desempeño de estas estrategias es muy pobre en redes reales, al aplicar en la red GoalBit (introducida en el Capítulo 2) ambas estrategias. Esto motiva la investigación de la concepción de nuevas estrategias de selección de piezas, que logren altas continuidades, manteniendo a su vez bajas latencias.

Se muestra un primer diseño de estrategia Golosa Ponderada, en la que se puede extender una consulta incluso si el par solicitante no posee la pieza y el solicitado sí. Matemáticamente se refutó la aplicación de esta estrategia, mostrando que la continuidad en la reproducción está acotada superiormente por valores inaceptables para cualquier instancia (dada por la terna M , N y q).

Finalmente se presenta la estrategia Mixta, que procura capturar las bondades de las estrategias clásicas. Si bien la estrategia Mixta muestra ventajas en su aplicación respecto de las clásicas estrategias, el problema es de tal amplitud que deja margen de mejoras. Un enfoque distinto, que logra mejorar las anteriores estrategias, se detalla en el siguiente capítulo. Allí se definen

medidas de optimalidad en base a estos parámetros y una familia de estrategias de selección de piezas con interesante riqueza y propiedades. El diseño se enfoca mediante un problema de optimización combinatoria, cuya complejidad motiva a proponer algoritmos de resolución aproximados.

Capítulo 4

Nuevas Estrategias de Selección de Piezas

4.1. Introducción

Este capítulo introduce un nuevo enfoque basado en el diseño de estrategias de permutación para resolver el modelo matemático presentado en la Sección 3.2 del Capítulo 3. En el mismo se ha hallado una ecuación que permite dada una estrategia de selección de piezas s_i , hallar el vector de probabilidades de ocupación p_i y consecuentemente la continuidad en la reproducción $c = p_N$ y el tiempo de buffering o latencia: $L = \sum_{i=1}^N p_i$.

El capítulo contiene 7 secciones. La Sección 4.2 define una familia de estrategias de selección de piezas. La esencia de esta familia de estrategias es determinar el orden de una consulta entre pares mediante una permutación arbitraria de los índices del buffer, disponiendo así de una estrategia por cada una de las $(N - 1)!$ permutaciones. Daremos la motivación de su definición, revisando varias propiedades destacadas de esta familia. De aquí en adelante se busca dentro de esta amplia familia aquellas estrategias con mayor desempeño, presentando la primera experiencia con el problema y un algoritmo primitivo de búsqueda orientado a lograr bajas latencias y altas continuidades.

El concepto de optimalidad se discute y formaliza en la Sección 4.3. Aquí se definen medidas de optimalidad aplicables a cualquier estrategia de selección, y se encuentra un nexo estrecho entre ellas cuando las soluciones son de buena calidad.

La Sección 4.4 formaliza el problema de búsqueda de estrategias de permutación a partir de un problema de optimización combinatoria (COP). Se revisan aquí las metaheurísticas conceptualmente valiosas para la comprensión de dos algoritmos propuestos para la resolución del problema.

La Sección 4.5 desarrolla el primero de dos algoritmos propuestos para la resolución del COP definido en la Sección 4.4. Introduce una métrica en el espacio de permutaciones, y una

noción de estructura de vecindad. Estos elementos permiten caracterizar a una búsqueda local en el espacio de estrategias de permutación, y ya este algoritmo garantiza retornar una permutación que es óptima localmente independientemente de la medida de optimalidad, siempre que el número de iteraciones en la búsqueda local no limite.

La Sección 4.6 brinda una propuesta más elaborada para la resolución del COP introducida. Veremos una reconfiguración del COP en términos del conocido problema del vendedor ambulante, o TSP (Travelling Salesman Problem [55, 87]). Para la resolución del TSP se aplica la metaheurística inspirada en el comportamiento de las hormigas para hallar las rutas más cortas, y conocida por sus siglas en inglés como ACO (Ant Colony Optimization [22, 38, 40, 41]), mezclada con una posterior búsqueda local, cuyas bondades se concen del primer algoritmo de esta sección. Una biyección existente entre ciclos en TSP y permutaciones en el COP permiten interpretar los resultados y definir nuevas estrategias de selección de piezas. Los resultados finales no garantizan la optimalidad global. Sin embargo, veremos empíricamente la obtención de soluciones de muy buena calidad, en relación a previas estrategias de selección de piezas, como lo son Greedy, Rarest First y la estrategia Mixta.

Por último, la Sección 4.7 contiene las conclusiones.

4.2. Familia de Permutaciones

En el Capítulo 3 hemos estudiado un modelo matemático que deja abierta la propuesta de realización de estrategias óptimas de selección de piezas. Más específicamente, si s_i representa la estrategia de selección de piezas, y p_i el vector de probabilidades de ocupación del buffer, sabemos que:

$$p_1 = \frac{1}{M} \quad (4.1)$$

$$p_{i+1} = p_i + (1 - p_i)p_i s_i. \quad (4.2)$$

A su vez, hemos visto propiedades de dos estrategias de selección de piezas a las que denominamos clásicas (remitimos por detalles al Capítulo 3). Una consiste en realizar la consulta de las piezas iniciando por los espacios del buffer cercanos a la línea de reproducción, llamada golosa o Greedy. Otra estrategia es la que se aplica en el protocolo BitTorrent y se llama Rarest First, solicitando las piezas más raras (más lejanas de la línea de reproducción), procurando así uniformizar la disponibilidad de piezas en la red. Las dos críticas más fuertes a estas estrategias clásicas son que la primera presenta frecuentemente bajos valores de continuidad, mientras que la segunda latencias inaceptables, para el modelo de streaming aquí expuesto. Más aún, veremos en el Capítulo 5 que el desempeño de estas estrategias es muy pobre en redes reales, al aplicar en la red GoalBit (introducida en el Capítulo 2) ambas estrategias. Estos elementos impulsan a diseñar nuevas estrategias de selección de piezas con mejores características, que es el objetivo de este capítulo.

Una manera natural de lograr una diversidad de opciones de estrategias, es considerar una permutación arbitraria de tamaño $N - 1$, y regirse por la misma para decidir el orden de consultas.

Denotaremos una permutación π formalmente como una aplicación biyectiva de los primeros $N - 1$ enteros positivos (índices del buffer):

$$\begin{aligned} \pi : \{1, \dots, N - 1\} &\rightarrow \{1, \dots, N - 1\}, \\ \forall i \neq j, \pi(i) &\neq \pi(j), \end{aligned}$$

donde N es el tamaño de buffer y $\pi(j)$ la evaluación de la permutación en j .

Definición 4.2.1 *La Familia de estrategias de permutación es el conjunto de estrategias cuyos órdenes de consulta se rigen por una permutación π arbitraria. Para cada permutación π , la estrategia correspondiente admite la siguiente expresión:*

$$s_{\pi(i)} = \left(1 - \frac{1}{M}\right) \prod_{j=1}^{i-1} (p_{\pi(j)} + (1 - p_{\pi(j)})^2) \quad (4.3)$$

La interpretación es similar a las clásicas, exceptuando que el orden de consulta es distinto. Primero se revisa el buffer en la posición $\pi(1)$. Si este par posee la pieza en la posición del buffer $\pi(1)$ (con probabilidad $p_{\pi(1)}$) o bien no la posee pero el par consultado tampoco (evento con probabilidad $(1 - p_{\pi(1)})^2$), se sigue mediante la revisión del buffer en la posición $\pi(2)$, y así sucesivamente.

Esta familia de permutaciones goza de propiedades muy interesantes, que nos guiarán a definir nuevos algoritmos que permitan superar el desempeño de estrategias Rarest First, Greedy, y Mixta, vistas en el Capítulo 3. Presentaremos tres algoritmos en orden creciente en bondades y complejidad en diseño. El primero se introduce en esta sección, y aprovecha el potencial de la familia de permutaciones definida en 4.2.1 para aproximar vectores de probabilidades de ocupación p . La primera experiencia de la búsqueda se orienta entonces a ingresar en este algoritmo vectores $p_i, i = 1, \dots, N$ cuya continuidad p_N sea la más alta posible, y latencia $L = \sum_{i=1}^N p_i$ sea baja.

4.2.1. Propiedades Básicas de la Familia

Propiedad 4.2.2 *Las estrategias clásicas y la Mixta son estrategias de permutación.*

Prueba. Rarest First se corresponde con la permutación identidad $\pi(i) = i, \forall i = 1, \dots, N - 1$, mientras que $\pi(i) = N - i, \forall i = 1, \dots, N - 1$ define a Greedy. Esto prueba que las estrategias clásicas son estrategias de permutación. Por último, que la estrategia Mixta toma un entero positivo $m : 1 < m < N$ y aplica el mismo orden de consulta que Rarest First en el espacio de buffer $[1, m]$ y el orden de Greedy en el sector restante $[m + 1, N - 1]$. Luego, para cada $m : 1 < m < N$ basta con elegir:

$$\pi(i) = i, \quad i = 1, \dots, m; \quad (4.4)$$

$$\pi(i) = N - (i - m), \quad i = m + 1, \dots, N - 1. \quad (4.5)$$

QED

La Propiedad 4.2.2 es la primera justificación de la búsqueda de estrategias de selección de piezas dentro de la familia de permutaciones. Asegura que la calidad del óptimo de las estrategias que se encuentren va a ser no menor que la de las estrategias clásicas y Mixta. Las siguientes dos propiedades evidencian la riqueza de esta nueva familia de estrategias.

Propiedad 4.2.3 *Sea π una permutación arbitraria, y s_i su correspondiente estrategia de selección. Luego, la secuencia $s_{\pi(i)}$ es monótona decreciente estrictamente.*

Prueba. Por su definición, todas las estrategias de permutación cumplen la siguiente recursión:

$$s_{\pi(1)} = 1 - \frac{1}{M} \quad (4.6)$$

$$s_{\pi(i+1)} = s_{\pi(i)}(p_{\pi(i)} + (1 - p_{\pi(i)})^2), \quad \forall i \in \{1, \dots, N - 2\} \quad (4.7)$$

Más aún, son no-negativas, por ser obtenidas mediante producto y sumas de factores y términos no-negativos. Puesto que $p_N < 1$, sabemos que $p_i \in [\frac{1}{M}, 1)$, $\forall i = 1, \dots, N$. Entonces, obtenemos la siguiente relación:

$$s_{\pi(i+1)} - s_{\pi(i)} = s_{\pi(i)}(p_{\pi(i)}(p_{\pi(i)} - 1)) < 0, \quad (4.8)$$

$$\forall i \in \{1, \dots, N - 2\}.$$

QED

Definición 4.2.4 *Dada una permutación arbitraria π de $\{1, \dots, N - 1\}$, el Sistema $S(\pi)$ consiste de las $N - 2$ ecuaciones definidas en la recurrencia (4.6, 4.7), las $N - 1$ ecuaciones (4.1) que definen el modelo y las incógnitas $\{p_{i=2, \dots, N}\} \cup \{s_{i=1, \dots, N-1}\} - \{s_{\pi(1)}\}$:*

$$p_1 = \frac{1}{M} \quad (4.9)$$

$$p_{i+1} = p_i + (1 - p_i)p_i s_i, \quad \forall i = 1, \dots, N - 1 \quad (4.10)$$

$$s_{\pi_1} = 1 - \frac{1}{M} \quad (4.11)$$

$$s_{\pi(i+1)} = s_{\pi(i)}(p_{\pi(i)} + (1 - p_{\pi(i)})^2) \quad \forall i = 1, \dots, N - 2 \quad (4.12)$$

Para cada permutación, es posible evaluar la performance (continuidad y latencia) resolviendo el sistema no lineal de ecuaciones $S(\pi)$.

Consideremos el Procedimiento 1, que construye una permutación dada una secuencia inyectiva arbitraria x_i .

La riqueza de esta familia de permutaciones se evidencia en la siguiente propiedad. Valiéndonos ahora de la monotonía de $s_{\pi(i)}$ (Lema 4.2.3) veremos que es posible aproximar cualquier secuencia inyectiva x_1, \dots, x_{N-1} mediante la estrategia cuya permutación asociada se construye en el Procedimiento 1, tal como se demuestra a continuación.

Propiedad 4.2.5 *“Propiedad de Aproximación de Estrategias”*

Para cada secuencia inyectiva x_1, x_2, \dots, x_{N-1} , existe un miembro de la familia de estrategias de permutación s tal que $\forall i, j : x_i > x_j \rightarrow s_i > s_j$.

Procedimiento 1 Construcción de Permutación de Aproximación**Entrada:** $x_i, i = 1, \dots, N - 1$ **Salida:** π

- 1: $I = \{1, \dots, N - 1\}$
- 2: **Para** $i = 1$ a $N - 1$
- 3: $j := \arg \max_{s \in I} x_s$
- 4: $\pi(i) := j$;
- 5: $I := I - \{j\}$;
- 6: **Fin Para**
- 7: **Retornar** π .

Prueba. Sea $x_i, i = 1 \dots N - 1$ una secuencia inyectiva arbitraria. Luego la permutación π construida en el Procedimiento 1 cumple por su definición que

$$x_{\pi(1)} > x_{\pi(2)} > \dots > x_{\pi(N-1)} \quad (4.13)$$

Llamemos s a la estrategia correspondiente a la permutación π . Entonces, por la Propiedad 4.2.3 se obtiene que:

$$s_{\pi(1)} > s_{\pi(2)} > \dots > s_{\pi(N-1)} \quad (4.14)$$

La tesis es evidente a partir de la observación de las expresiones (4.13) y (4.14).

QED

Nótese que la Propiedad de Aproximación permite aproximar la forma de cualquier estrategia factible, siempre que sea inyectiva. Todas las estrategias de permutación son claramente inyectivas, y hasta la fecha se desconocen ejemplos de estrategias factibles y no inyectivas, por lo que la hipótesis no es restrictiva. Definiendo la estrategia de permutación del modo anterior asegura que el comportamiento de crecimiento-decrecimiento de cada estrategia pueda ser aproximado, por una correspondiente estrategia de permutación.

4.2.2. Experiencia en la Búsqueda de soluciones

La experiencia comprueba que un estudio exhaustivo de todas las estrategias de permutación puede ser realizada solamente para tamaños de buffer limitados, ya que el esfuerzo computacional resulta ser prohibitivo. Consideremos las 4 medidas de optimalidad siguientes:

- 1) En continuidad: aquella estrategia que logra la mejor continuidad.
- 2) En latencia: la estrategia que logra la latencia más baja.
- 3) Optimalidad minimax: entre las estrategias con mejor continuidad que Greedy, la de menor latencia.
- 4) Optimalidad maximin: entre las estrategias con menor latencia que Rarest First, la de mejor continuidad.

Un estudio completo de la performance para tamaños de buffer limitados entre $N = 4$ y $N = 9$ y 5 pares se puede encontrar en el Cuadro 4.1. Mientras que Greedy en todos estos casos domina en cuanto a latencia, existen ciertas permutaciones que alcanzan la mejor continuidad. Además, estas estrategias presentan un pico cerca del centro del buffer. Esta es una propiedad que guía la metodología de diseño en esta sección.

	OPT	PERMUTACION	CONTINUIDAD	LATENCIA
N=4	1	123	0.61	1.61
	2	321	0.59	1.47
	3	123	0.61	1.61
	4	321	0.59	1.47
N=5	1	1423	0.70	2.21
	2	4321	0.69	1.98
	3	1432	0.70	2.18
	4	4321	0.69	1.98
N=6	1	15243	0.77	2.87
	2	54321	0.75	2.48
	3	52413	0.76	2.61
	4	54321	0.75	2.48
N=7	1	162543	0.83	3.56
	2	654321	0.81	2.97
	3	654321	0.81	2.97
	4	654321	0.81	2.97
N=8	1	7126534	0.86	4.05
	2	7654321	0.85	3.45
	3	7654321	0.85	3.45
	4	7654321	0.85	3.45
N=9	1	81723645	0.89	4.68
	2	87654321	0.88	3.91
	3	87654321	0.88	3.91
	4	87654321	0.88	3.91

Cuadro 4.1: El estudio exhaustivo es solamente posible para tamaños limitados de buffer. A efectos ilustrativos se muestra aquí las permutaciones que logran mejor continuidad y mejor latencia según los 4 tipos de optimalidad mencionados, para el caso de 5 pares y tamaño de buffer variable entre 4 y 9.

El Algoritmo 2 recibe una probabilidad de ocupación deseada p (entrada a diseñar en la línea 1) y retorna una permutación π . La finalidad de este algoritmo es buscar la permutación π tal que su correspondiente estrategia de permutación s logre probabilidades de ocupación lo más similares a p que sea posible. Este es el primer paso para automatizar la búsqueda de estrategias de permutación con buena calidad (alta continuidad y baja latencia). Su principal potencial es el de explotar la Propiedad de Aproximación de Estrategias. El paso 1 consiste en elegir una secuencia $p_i, i = 1, \dots, N - 1$ a la entrada, que posea buenas características. Por ahora la medida “alta continuidad” refiere a continuidad superior a aquella de Rarest First, y la

latencia comparable con Greedy. Se calcula en las líneas 2 y 3 la calidad de la probabilidad de ocupación p antes diseñada. En las líneas 4 a 6 se halla la estrategia s_{ideal} que idealmente lograría la probabilidad de ocupación p . Se utiliza a continuación la Propiedad de Aproximación de Estrategias: se aproxima a s_{ideal} mediante s , que es la estrategia asociada a la permutación salida del Procedimiento 1. En la línea 8 se resuelve el sistema no lineal $S(\pi)$ mediante el método de Newton-Raphson, para poder obtener la estrategia de permutación s y su correspondiente vector de probabilidades de ocupación denotado con p^* ¹. La calidad de la estrategia así obtenida se calcula en las líneas 9 y 10. Por último, en caso de alejarse un 10 por ciento respecto de la continuidad y latencia de la p diseñada de la línea 1, se debe volver a la línea 1 y rediseñar la probabilidad de ocupación p . En caso contrario la aproximación se considera aceptable: se opta por regir el orden de consulta mediante la permutación π , que se retorna en la salida. Este algoritmo tiene la ventaja de aproximar una probabilidad de ocupación deseada. Por contrapartida, requiere diseñar la entrada apropiadamente, como veremos a continuación.

Algoritmo 2 Aplica el Procedimiento 1 para encontrar la estrategia de permutación más próxima a la que obtendría idealmente una probabilidad de ocupación de entrada.

- 1: Diseñar una secuencia $p_i, i = 1, \dots, N$.
 - 2: $c_{in} = p_N$
 - 3: $L_{in} = \sum_{i=1}^N p_i$
 - 4: **Para** $i = 1$ a $N - 1$
 - 5: $s_{ideal}(i) = \frac{p_{i+1} - p_i}{(1 - p_i)p_i}, i = 1, \dots, N - 1$. // Estrategia que logra p .
 - 6: **Fin Para**
 - 7: $\pi \leftarrow \text{Procedimiento1}(s_{ideal})$
 - 8: $(s, p^*) \leftarrow \text{ResolverSistema}S(\pi)$
 - 9: $c = p_N^*$
 - 10: $L = \sum_{j=1}^N p_j^*$
 - 11: **Si** $c < 0,9c_{in}$ o $L_{in} < 0,9L$ **entonces**
 - 12: **Retornar al Paso 1**
 - 13: **Fin Si**
 - 14: **Retornar** π
-

El Algoritmo 2 es aplicado a diversas secuencias p . Un diseño ambicioso de la entrada es una secuencia exponencial con continuidad de uno, por su regularidad y bajos valores de latencia, es decir:

$$p_{\epsilon_i} = M^{\frac{N-i}{1-N}} \quad i = 1, \dots, N.$$

¹En caso de existir una norma menor que 1 para el jacobiano no singular del sistema $S(\pi)$ en todo su dominio (que es un cubo; por tanto compacto y convexo), el método de Newton-Raphson para sistemas asegura orden de convergencia cuadrática. Si bien la aplicación de Newton-Raphson a este sistema no lineal resultó en todos los casos convergente, la existencia de tal norma es un problema abierto.

Sin embargo, con tal vector de entrada utilizando que $p_{\epsilon_{i+1}} = M^{\frac{1}{N-1}} p_{\epsilon_i} > p_{\epsilon_i}$, la correspondiente estrategia s_{ideal} obtenida en el bloque de líneas 4 a 6 verifica que:

$$\frac{s_{ideal}(i+1)}{s_{ideal}(i)} = \frac{p_{\epsilon_{i+2}} - p_{\epsilon_{i+1}}}{(1 - p_{\epsilon_{i+1}})p_{\epsilon_{i+1}}} \frac{(1 - p_{\epsilon_i})p_{\epsilon_i}}{p_{\epsilon_{i+1}} - p_{\epsilon_i}} = \frac{1 - p_{\epsilon_i}}{1 - p_{\epsilon_{i+1}}} > 1, \quad \forall i \in \{1, \dots, N-2\}. \quad (4.15)$$

Esto implica que la estrategia a simular es monótona creciente estricta, y la permutación que se construye en la línea 7 es $\pi(i) = N - i$. El algoritmo cae en la estrategia golosa, algo no deseable.

Ingresamos ahora al Algoritmo 2 una probabilidad de ocupación segmentada a tramos mostrada en la Figura 4.1 para el caso $M = 1000$, $N = 40$. La estrategia ideal correspondiente no es realizable empíricamente, pues la magnitud de s_{ideal} supera 1 en algunos índices, y por tanto no representa probabilidades. No obstante, este diseño de la entrada brinda un buen panorama del funcionamiento de este algoritmo. Las funciones p , s_{ideal} , s y p^* correspondientes a la aplicación de p segmentada en el Algoritmo 2 se aprecian en las Figuras 4.1, 4.2, 4.3 y 4.4 respectivamente.

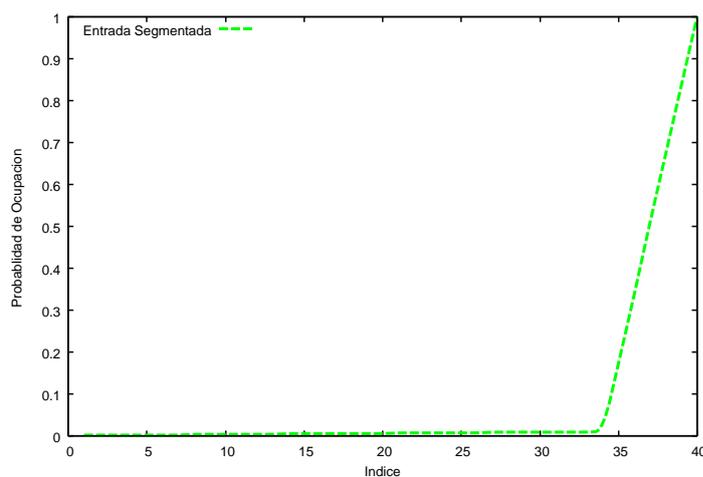


Figura 4.1: Probabilidad de ocupación segmentada

Se puede observar que un pico en la estrategia de permutación (ver Figura 4.3) genera un cambio abrupto en la probabilidad de ocupación (cambio de pendientes en Figura 4.4). El lugar donde se halla el pico en la estrategia de permutación nos permite elegir donde estará el cambio de comportamiento de la probabilidad de ocupación. La posición del pico en la estrategia de permutación es crítica, pues si se coloca muy cerca de la línea de reproducción, se tendrá una baja continuidad. En caso contrario, si el pico se dispone muy lejos de la reproducción, los valores altos de probabilidad de ocupación desde el mismo acarrearán como resultado altos valores de latencia.

Definición 4.2.6 La subfamilia de permutaciones son los miembros de la familia de permutaciones cuya estrategia de selección presenta exactamente un máximo relativo (pico), que no es

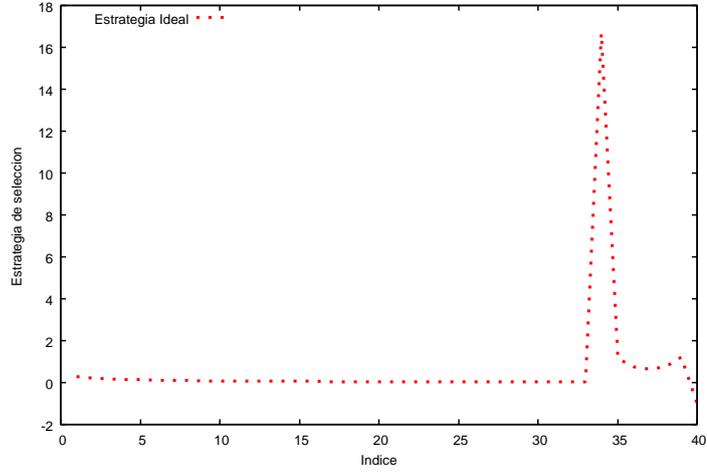


Figura 4.2: Estrategia ideal para la probabilidad de ocupación segmentada

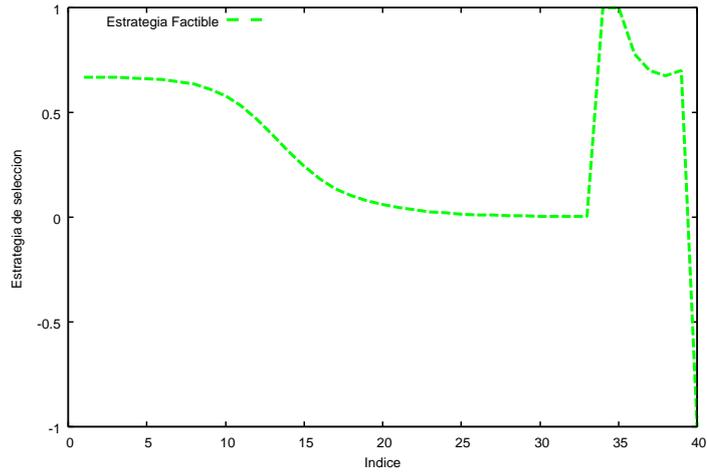


Figura 4.3: Estrategia de permutación para el objetivo segmentado

absoluto. Para cada par de naturales $(I, J) : I + J < N$, hay una permutación de la subfamilia que se puede expresar de la siguiente manera:

$$\pi(i) = N - i, i = 1, \dots, I, \quad (4.16)$$

$$\pi(I + j) = j, j = 1, \dots, J \quad (4.17)$$

$$\pi(I + J + k) = \left\lfloor \frac{N + J - I}{2} \right\rfloor + \left\lceil \frac{k}{2} \right\rceil (-1)^{k+1}, \quad (4.18)$$

$$k = 1, \dots, N - I - J - 1.$$

Analicemos ahora la subfamilia introducida en la Definición 4.2.6. El pico de los miembros de la subfamilia se reduce en magnitud a medida que aumentan los parámetros de control

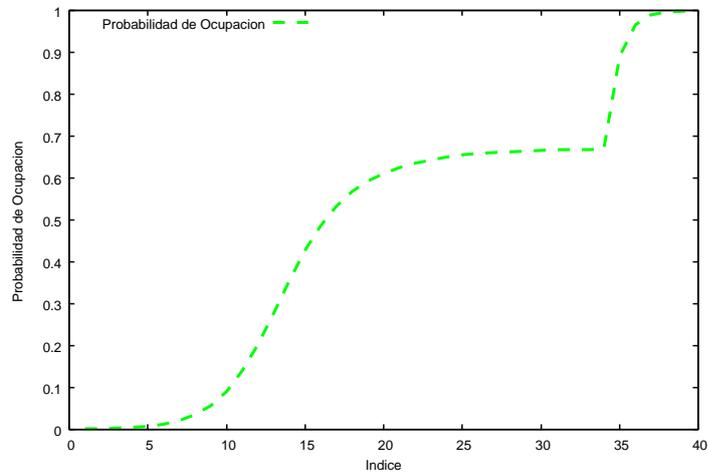


Figura 4.4: Probabilidad de ocupación en la salida, asociada a la entrada segmentada

I y J (índices en los extremos). Esto trae como consecuencia un efecto de reducción de la latencia. Además, las primeras solicitudes en los extremos del buffer ayudan a incrementar la continuidad, controlando a la vez la latencia.

Las Figuras 4.5 y 4.6 muestran la evolución de la continuidad y latencia para diferentes pares I, J variando entre 0 y 7, y diferentes números de pares y tamaños de buffer. Se puede apreciar que para $N = 40$, la continuidad es prácticamente de 1, salvo unos pocos casos particulares. La latencia incrementa proporcionalmente con el tamaño de buffer. Es importante notar que la latencia se puede mantener baja, incluso con un número elevado de pares. Esta subfamilia introducida en la Definición 4.2.6 logra escalabilidad.

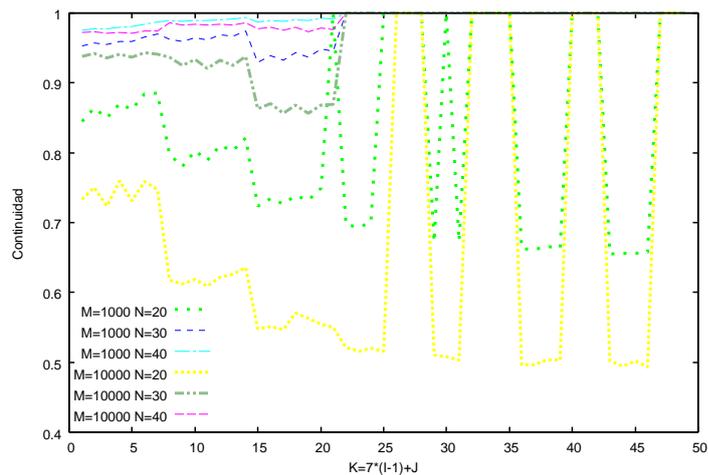


Figura 4.5: Evolución de la continuidad

Para la subfamilia definida en 4.2.6, una alta continuidad de 0.989 junto con una latencia relativamente baja de 9.03 puede ser obtenida en el caso de $M = 1000$ pares y tamaño de

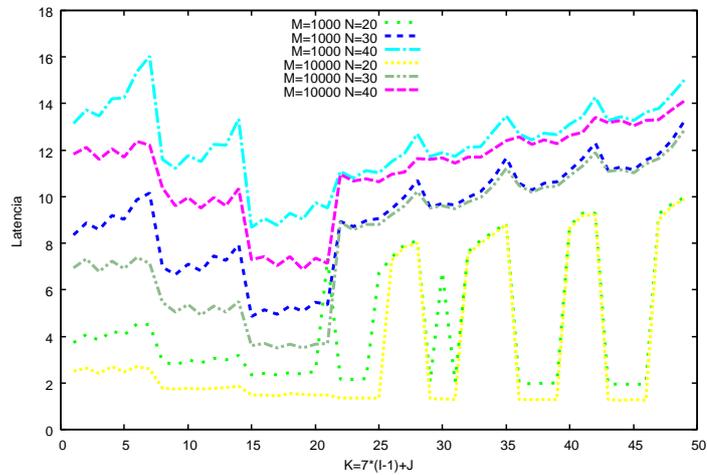


Figura 4.6: Evolución de la latencia

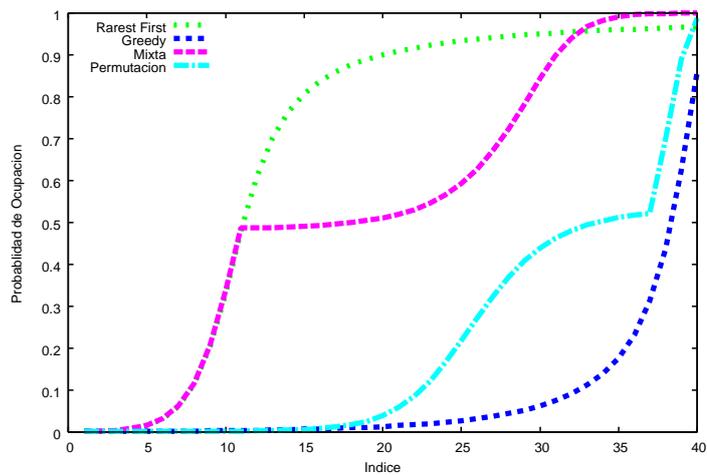


Figura 4.7: Probabilidad de ocupación para cada estrategia

buffer de $N = 40$, con $I = 3$ y $J = 1$ ($K = 15$ en las gráficas). La Figura 4.7 presenta las probabilidades de ocupación correspondientes a las estrategias Greedy, Rarest First, Mixta y la estrategia de permutación propuesta para este caso. La estrategia de permutación propuesta es el elemento de la subfamilia con $I = 3$ y $J = 1$. El desempeño resultante se resume en el Cuadro 4.2.

Mientras que la continuidad de la permutación propuesta es prácticamente la que se alcanza con la estrategia Mixta, su latencia es sustancialmente menor, como se puede apreciar en la Figuras 4.8 y 4.9.

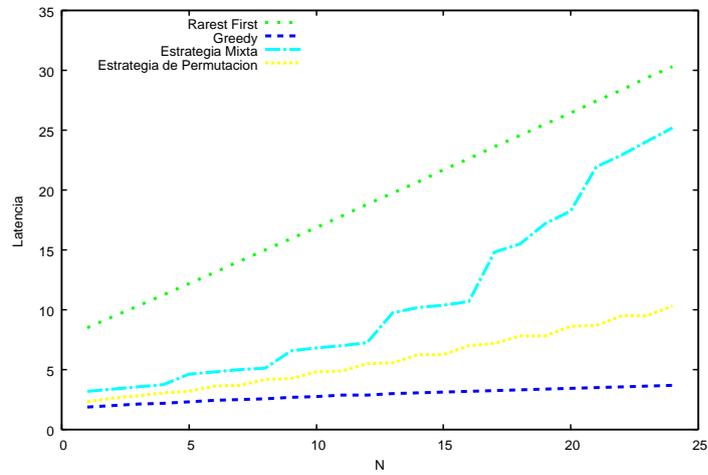


Figura 4.8: Comparación de la Continuidad entre diferentes estrategias

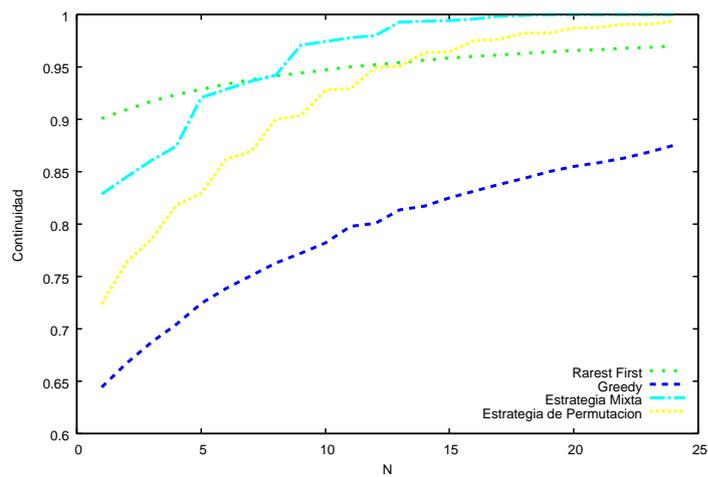


Figura 4.9: Comparación de la Latencia entre diferentes estrategias

Estrategia	latencia	Continuidad
Rarest First	27.43	0.967
Greedy	3.49	0.859
Mixta	21.9	0.999
Permutación	9.03	0.989

Cuadro 4.2: Comparación de la performance de distintas estrategias.

4.3. Medidas de Optimalidad

Daremos a continuación medidas de optimalidad inspiradas en el mejor vector de probabilidades de ocupación p_i^* , el cual debe tomar valores pequeños en índices del buffer menores

que N (a los efectos de tener baja latencia) y al mismo tiempo el valor de continuidad en la reproducción p_N lo más alto posible.

4.3.1. Medida basada en la extensión de una consulta

Consideremos una permutación arbitraria π de los elementos $\{1, \dots, N-1\}$. Toda consulta se realiza en una cantidad finita de pasos, donde cada paso consiste en el chequeo de una posición del buffer. En particular, toda consulta exitosa logra obtener una pieza en no más de $N-1$ pasos. Sea $\alpha_i = p_{\pi(i)}(1 - p_{\pi(i)})$ la probabilidad de beneficiarse con una pieza consultando solamente la posición del buffer $\pi(i)$, y X_π la variable aleatoria que cuenta la extensión (cantidad de revisiones del buffer) de una consulta exitosa, utilizando la estrategia de permutación π . Esta variable aleatoria tiene características similares a una Geométrica, con la salvedad que sus probabilidades de éxito cambian en cada lugar del buffer.

Proposición 4.3.1 *El número esperado de pasos necesarios en una consulta exitosa mediante la estrategia s_π es:*

$$E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} i(p_{\pi(i)+1} - p_{\pi(i)}). \quad (4.19)$$

Prueba. Sea X_π la variable aleatoria que cuenta el número de consultas necesarias para obtener una pieza usando la estrategia π . Entonces:

$$\begin{aligned} E(X_\pi) &= \sum_{i=1}^{N-1} i[p_{\pi(i)}(1 - p_{\pi(i)})] \prod_{j=1}^{i-1} (p_{\pi(j)} + (1 - p_{\pi(j)})^2) \\ &= \frac{M}{M-1} \sum_{i=1}^{N-1} i p_{\pi(i)} (1 - p_i) s_{\pi(i)} \\ &= \frac{M}{M-1} \sum_{i=1}^{N-1} i (p_{\pi(i)+1} - p_{\pi(i)}) \end{aligned}$$

QED

El valor medio de la extensión de la consulta es una combinación lineal de la sumatoria de saltos en todos los índices consecutivos de la probabilidad de ocupación. Nótese en particular que esta esperanza incrementa a medida que aumenta la continuidad p_N , pues el índice que hace que $\pi_i = N-1$ es tal que $p_{\pi_i+1} = p_N$, y nunca es cancelado.

Para el caso de la estrategia de selección Rarest First, se tiene que la esperanza de esta V.A. X_π se reduce a:

$$E(X_\pi) = \frac{M}{M-1} (Np_N - L) \quad (4.20)$$

Donde L es la latencia inicial del video (o tiempo de buffering) medida en ranuras de tiempo, y p_N es la continuidad en la reproducción del video. Esto explica que hay un vínculo estrecho en los parámetros de calidad y la extensión media de una consulta. Una pregunta pertinente aquí es la siguiente: ¿deseamos que las consultas sean extensas o cortas? Para responderla debemos recordar la hipótesis de que todo el mecanismo de consulta tiene una duración inferior a una ranura de tiempo, por lo que la extensión no preocupa en cuanto a tiempos refiere. Más aún, una consulta extensa significa que tanto el par consultado como el consultante poseen una buena cantidad de piezas en su buffer. Esto indica que es deseable un valor esperado de largo de consultas mayor.

La expresión anterior motiva entonces a nuevas medidas de optimalidad. Consideremos la medida de calidad motivada de Rarest First: $M_0 = Np_N - L$.

Nota: 4.3.2 Si damos prioridad a la latencia, es interesante notar que maximizar M_0/L resulta equivalente a maximizar la “calidad cociente”: $M_1 = p_N/L$, puesto que

$$\frac{M_0}{L} = \frac{Np_N}{L} - 1 = NM_1 - 1, \quad (4.21)$$

a partir de la igualdad (4.21), dado un tamaño de buffer N fijo es claro que maximizando M_1 estamos maximizando $\frac{M_0}{L}$, y recíprocamente.

Es interesante analizar el caso de Greedy, en el cual la expresión para $E(X_\pi)$ también se simplifica:

$$E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} i(p_{N-i+1} - p_{N-i}) \quad (4.22)$$

$$= \frac{M}{M-1} \sum_{j=1}^{N-1} (N-j)(p_{j+1} - p_j) \quad (4.23)$$

$$= \frac{M}{M-1} (L - Np_1) \quad (4.24)$$

Otra pregunta que cautiva la atención: ¿cuál de las 2 estrategias clásicas posee menor extensión media de una consulta? Si bien este problema está abierto, daremos una nueva interpretación del valor esperado de la extensión de una consulta para Rarest First y Greedy. Para Rarest First, es posible reescribir la expresión (4.20) de la siguiente manera:

$$E(X_\pi) = \frac{M}{M-1} (Np_N - L) = \frac{M}{M-1} \sum_{i=1}^N (p_N - p_i). \quad (4.25)$$

Por su parte, para Greedy se puede reescribir (4.22) así:

$$E(X_\pi) = \frac{M}{M-1} (L - Np_1) = \frac{M}{M-1} \sum_{i=1}^N (p_i - p_1) \quad (4.26)$$

En Rarest First, la igualdad (4.25) establece que el número esperado de pasos de una consulta en Rarest First es proporcional a la suma global de las distancias entre todos los índices del vector p y su continuidad. Se resalta el interés de mantener alto este valor, que implica directamente bajas latencias (parámetro débil de Rarest First). Por otra parte, en Greedy la extensión esperada de una consulta es proporcional a la suma global de diferencias entre el vector de probabilidad de ocupación p_i y el valor base p_1 . Esto mide la manera en que Greedy se aparta del valor base p_1 . Conocidas ya las bondades de baja latencia en Greedy, este apartamiento favorece a aumentar la continuidad, hecho que se justifica por la monotonía de p .

4.3.2. Dos problemas de Optimización

Se hace necesario luego de este análisis capturar mediante una medida de optimalidad a ambos parámetros de desempeño. En todos los casos deseamos altos valores de continuidad p_N y reducidas magnitudes de la probabilidad de ocupación en el resto de los índices $\{1, \dots, N - 1\}$.

Definición 4.3.3 *El vector probabilidad de ocupación ideal es p^* :*

$$\begin{aligned} p_i^* &= 1/M \quad \forall i = 1 \dots N - 1 \\ p_N^* &= 1. \end{aligned} \quad (4.27)$$

La simultánea maximización de continuidad y minimización de latencia se puede capturar en los siguientes modelos de programación matemática, sujetos a las mismas ligaduras:

Problema 1

$$\begin{aligned} \max M_1 &= \frac{p_N}{\sum_{i=1}^N p_i} \\ \text{s.a.} \\ p_1 &= \frac{1}{M} \\ 1 &\geq p_N > c \\ p_i &\leq p_{i+1}, \quad \forall i = 1, \dots, N - 1 \end{aligned}$$

Problema 2

$$\begin{aligned} \min ECM(p, p^*) &= \frac{1}{N} \sum_{i=1}^N (p_i - p_i^*)^2 \\ \text{s.a.} \\ p_1 &= \frac{1}{M} \\ 1 &\geq p_N > c \\ p_i &\leq p_{i+1}, \quad \forall i = 1, \dots, N - 1 \end{aligned}$$

El primer problema es una propuesta simple y natural: busca maximizar el cociente entre la continuidad y latencia. También se sustenta en que este problema coincide con maximizar la extensión esperada de una consulta, normalizada por la latencia, en Rarest First. Posee las ventajas de tener clara cota superior para la función objetivo (que es 1), y además la misma es una función adimensionada. Luego de definir el vector p^* , que intuitivamente es la mejor probabilidad de ocupación, es de esperar considerar una función objetivo que mida una distancia con ésta, tal como se formula en el Problema 2 mediante la consideración del error cuadrático medio con esta probabilidad de ocupación ideal.

El lema siguiente confirma la optimalidad de p^* en los dos problemas de optimización recientemente propuestos. El segundo lema es pesimista, y establece que es imposible obtener esta probabilidad de ocupación prácticamente mediante alguna estrategia de permutación. Esto es muy importante, y motiva a buscar la probabilidad de ocupación factible que sea “la más próxima” a p^* . Más formalmente, buscar la permutación π tal que al calcular el vector p correspondiente (mediante resolución del Sistema $S(\pi)$), se logre el mayor valor de la función objetivo del Problema 1 o el menor del Problema 2.

Lema 4.3.4 *La probabilidad de ocupación p^* (Def. 4.3.3) es la secuencia óptima para ambos problemas de optimización.*

Prueba. Es evidente que p^* es óptima en el Problema 2, dado que su función objetivo es el cuadrado de la norma euclídea, multiplicada por una constante positiva, y p^* alcanza su cero, cumpliendo a la vez con las ligaduras. Para probar la optimalidad en el Problema 1, observemos que la función objetivo es monótona decreciente estricta con respecto a p_i , para cada índice $i < N$. Para minimizar cumpliendo con las ligaduras se debe elegir $p_i = \frac{1}{M}$, $\forall i < N$. Por otra parte, la función objetivo es monótona creciente con respecto a la continuidad, lo que implica que p_N debe tomarse lo mayor posible cumpliendo con las ligaduras, es decir: $p_N = 1$. La solución nuevamente resulta la probabilidad de ocupación ideal, p^* .

QED

Lema 4.3.5 *La probabilidad de ocupación ideal, p^* , no es realizable.*

Prueba. Sabemos que toda estrategia debe cumplir con la ecuación (4.1):

$$p_{i+1} = p_i + (1 - p_i)p_i s_i, \quad \forall i = 1, \dots, N - 1$$

En particular evaluando en $i = N - 1$, la estrategia asociada a p^* debe cumplir que:

$$s_{N-1} = \frac{p_N^* - p_{N-1}^*}{(1 - p_{N-1}^*)p_{N-1}^*} = \frac{1 - 1/M}{(1 - 1/M)1/M} = M > 1. \text{ Esto último es imposible, dado que } s_{N-1} \text{ es una probabilidad.}$$

QED

Este último resultado afirma que nos debemos conformar en todos los casos con estrategias que logran subóptimos, en cualquiera de las dos medidas de optimalidad consideradas.

4.3.3. Discusión sobre Optimalidad en ambos problemas

Primero que nada observemos que para ternas particulares (M, N, c) , no existen estrategias de permutación que cumplan con las restricciones de ambos problemas. Por ejemplo, si $M = N = 5$ la permutación que logra la mejor continuidad de $c = 0,7$ es $\pi = \{1, 4, 2, 3\}$ (Cuadro 4.1 con $N = 5$), por lo que en este caso no es posible lograr continuidades superiores a 0.7. Dado que es prohibitivo el esfuerzo computacional requerido para llevar a cabo una búsqueda exhaustiva para altos tamaños de buffer, este estudio se enfoca en obtener una medida de similitud de los Problemas 1 y 2, y cotas de calidad.

Lema 4.3.6 *Consideremos una calidad de cociente fija: $k = \frac{p_N}{\sum_{i=1}^N p_i}$. Entonces, la esfera tangente al previo hiperplano que está centrada en p^* tiene un radio $r(k)$ igual a:*

$$r(k) = \frac{1 - \frac{k}{k_{max}}}{\sqrt{(N - 2)k^2 + (1 - k)^2}}$$

$$\text{donde } k_{max} = \frac{M}{M + N - 1}.$$

Prueba. El radio $r(k)$ coincide con la magnitud del producto interno entre p^* y el versor normal del hiperplano $k = \frac{p_N}{\sum_{i=1}^N p_i}$. Este último admite la siguiente expresión implícita:

$$(k - 1)p_N + k \sum_{i=2}^{N-1} p_i = -\frac{k}{M}. \text{ Entonces, si llamamos } n_k \text{ al versor normal de este hiperplano:}$$

$$r(k) = |\langle p^*, n_k \rangle| = \left| \left\langle \left(\frac{1}{M}, \dots, \frac{1}{M}, 1 \right), \frac{(k, \dots, k, k - 1)}{\sqrt{(N - 2)k^2 + (1 - k)^2}} \right\rangle \right| = \frac{|k \frac{N-1}{M} + (k - 1)|}{\sqrt{(N - 2)k^2 + (1 - k)^2}} \quad (4.28)$$

Por el Lema 4.3.4, la calidad cociente es óptima en p^* . Si llamamos k_{max} a la respectiva calidad cociente máxima, esta se puede obtener mediante simple evaluación:

$$k_{max} = Q(p^*) = \frac{1}{\frac{N-1}{M} + 1} = \frac{M}{M + N - 1} \quad (4.29)$$

Por último, reescribiendo $r(k)$ y haciendo uso de la expresión (4.29) de calidad máxima, obtenemos:

$$r(k) = \frac{|k^{\frac{N+M-1}{M}} - 1|}{\sqrt{(N-2)k^2 + (1-k)^2}} = \frac{1 - \frac{k}{k_{max}}}{\sqrt{(N-2)k^2 + (1-k)^2}}, \quad (4.30)$$

QED

Definición 4.3.7 Dado un entero $N > 2$, la región factible $R \subseteq \mathcal{R}^N$ tiene dimensión $N - 1$ y es: $R = \{(p_1, \dots, p_N) : 1/M = p_1 \leq p_2 \leq \dots \leq p_N \leq 1\}$

Lema 4.3.8 Consideremos el caso $N = 3$. Dada una calidad de cociente fija $k = \frac{p_3}{p_1 + p_2 + p_3}$, el área que delimita este plano y la esfera centrada en p^* intersectado con la región factible, es, en proporción con el área de la región factible (ver Fig. 4.10):

$$A = \frac{1}{2(1 - \frac{1}{M})^2} (1 - \frac{k}{k_{max}})^2 (\frac{2}{k(1-k)} - \frac{\pi}{k^2 + (k-1)^2}), \text{ donde } k_{max} = \frac{M}{M+2}.$$

Prueba. La relación de áreas a calcular se puede determinar simplemente a partir de la diferencia entre el área de un triángulo y un cuarto de disco (ver Figura 4.10). Finalmente dividiendo entre el área de la región factible.

Por el Lema 4.3.6, el área del cuarto de disco es:

$$A_{D_4} = \frac{\pi r(k)^2}{4} = \pi \frac{(1 - \frac{k}{k_{max}})^2}{4(k^2 + (1-k)^2)} \quad (4.31)$$

Por otra parte, el área del triángulo delimitado por el plano de calidad constante y la región factible se halla mediante geometría elemental, y vale:

$$A_T = \frac{(1 - \frac{k}{k_{max}})^2}{2k(1-k)} \quad (4.32)$$

Por último, la región factible $R = \{(p_1, p_2, p_3) : p_1 = \frac{1}{M} \leq p_2 \leq p_3 \leq 1\}$ es un triángulo contenido en el plano $p_1 = \frac{1}{M}$, cuyos vértices son p^* (la solución ideal), $(\frac{1}{M}, \frac{1}{M}, 1)$ y $(\frac{1}{M}, 1, 1)$. Su área es igual a:

$$A_R = \frac{(1 - \frac{1}{M})^2}{2}. \quad (4.33)$$

En conclusión, la porción de área comprendida entre las dos regiones de calidad constante, referenciada respecto al área de la región factible, es:

$$A = \frac{A_T - A_{D_4}}{A_R} = \frac{1}{2(1 - \frac{1}{M})^2} (1 - \frac{k}{k_{max}})^2 (\frac{2}{k(1-k)} - \frac{\pi}{k^2 + (k-1)^2}). \quad (4.34)$$

QED

La última expresión se puede extender al caso N -dimensional. Obsérvese que la relación de áreas A del Lema 4.3.8 nos brinda una medida de no coincidencia de soluciones a los Problemas de optimización 1 y 2. Más aún, si asumimos que la disposición en el N -espacio de los $(N - 1)!$ vectores de probabilidad correspondientes a la familia de permutaciones se distribuye uniformemente, A mide la probabilidad de no coincidencia de las soluciones de los distintos problemas, para el caso $N = 3$. Generalicemos este concepto a continuación.

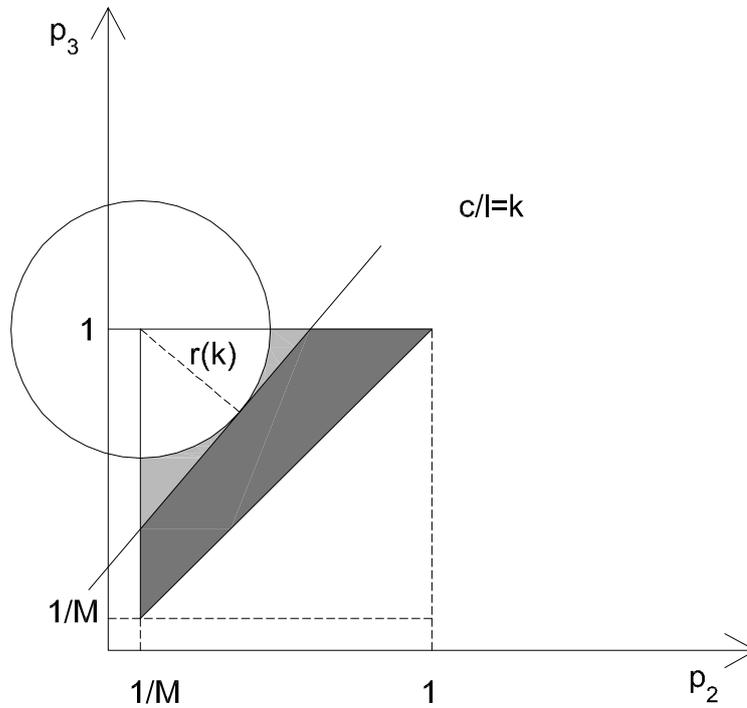


Figura 4.10: Representación gráfica de niveles de calidad constante para ambos problemas de programación matemática (plano $p_1 = \frac{1}{M}$)

Lema 4.3.9 El punto más distante a p^* contenido en la región factible y en el hiperplano de calidad cociente k es

$$q = \left(\frac{1}{M}, \dots, \frac{1}{M}, \frac{k}{1-k} \frac{N-1}{M} \right).$$

Prueba. Debemos resolver el Problema 2, pero su maximización (peor calidad) sujeta a la vez a la pertenencia al hiperplano de calidad cociente k . Consideremos la siguiente relajación lagrangeana, donde $d(p^*, p)$ es la distancia euclídea entre p y p^*

$$\begin{aligned} \min_p \phi(p_i, \lambda) = & -d^2(p, p^*) + \lambda_1 \left(\frac{1}{M} - p_2 \right) + \sum_{i=2}^{N-1} \lambda_i (p_i - p_{i-1}) + \\ & \lambda_N (p_N - 1) + \lambda_{N+1} \left(p_N - k \left(\frac{1}{M} + \sum_{i=2}^N p_i \right) \right) \\ \text{s.a.} & \\ & \lambda_i \geq 0, \quad \forall i = 1, \dots, N \end{aligned}$$

Apliquemos el Teorema Fundamental de relajación lagrangeana para hallar el punto deseado q . Calculemos en primera instancia las derivadas parciales del lagrangeano, respecto a las varia-

bles de decisión:

$$\frac{\partial \phi}{\partial p_i} = -2(p_i - \frac{1}{M}) - \lambda_{i+1} + \lambda_i - k\lambda_{N+1} = 0, \quad \forall i = 2, \dots, N-1 \quad (4.35)$$

$$\frac{\partial \phi}{\partial p_N} = -2(p_N - 1) - \lambda_{N-1} + \lambda_N + \lambda_{N+1}(1 - k) = 0. \quad (4.36)$$

Para hallar el óptimo del problema original se debe respetar la pertenencia a la región factible de la variable de decisión. Esto equivale a las siguientes restricciones:

$$p_2 \geq \frac{1}{M} \rightarrow -\lambda_2 + \lambda_1 - k\lambda_{N+1} \geq 0 \quad (4.37)$$

$$p_{i+1} \geq p_i \rightarrow -\lambda_{i+1} - 2\lambda_i + \lambda_{i-1} \geq 0, \quad \forall i = 2, \dots, N-1 \quad (4.38)$$

$$p_N \leq 1 \rightarrow \lambda_{N+1}(1 - k) \leq \lambda_{N-1} - \lambda_N \quad (4.39)$$

A estas restricciones se agregan la pertenencia al hiperplano y las condiciones de complementariedad. El punto q está en la región factible como también en el hiperplano. Aplicando el Teorema de relajación de Lagrange, basta con cumplir las condiciones de complementariedad y ser un óptimo del problema relajado (cumpliendo con la no negatividad de los multiplicadores, salvo posiblemente λ_{N+1}), para ser el óptimo del problema original. Resolviendo la recursión (4.38) bajo el caso de igualdades, sobre los multiplicadores se obtiene que $\lambda_n = \lambda_1 + k\lambda_{N+1}(1 - n)$, $\forall n < N$. La igualdad (4.36) implica que

$$\lambda_N = 2(p_N - 1) - \lambda_{N-1}(1 - k) + \lambda_{N+1}.$$

Finalmente se observa que tomando:

$$\lambda_1 = \text{máx} \{k\lambda_{N+1}(N - 2), 2(1 - p_N) + \lambda_{N+1}(1 - k(N - 1)), 0\}$$

todos los multiplicadores (excepto λ_{N+1}) resultan ser no negativos. Por último nótese que el gradiente del problema relajado se anula (sus derivadas parciales se expresan en las ecuaciones (4.35) y (4.36)), y este extremo es justamente un mínimo (el máximo absoluto se obtiene en p^* en el problema relajado). De esta manera, la optimalidad de q queda probada.

QED

Definición 4.3.10 Sea $N > 2$ un entero y $B(r, p^*) = \{p \in R^N : p_1 = 1/M, \|p - p^*\| \leq r\}$ la $N - 1$ -bola cerrada de radio r centrada en p^* . Considérese el semiespacio \mathcal{H} limitado por el hiperplano de calidad cociente $k(r)$ constante: $\mathcal{H} = \{p \in R^N : p_1 = 1/M, p_N / \sum_{i=1}^N p_i \geq k(r)\}$. Si $R_{eq}(r) = R \cap (\mathcal{H} - B(r, p^*))$ es la región de equivocación, se define la relación de equivocación $E(r)$ como:

$$E(r) = \frac{\text{Vol}(R_{eq}(r))}{\text{Vol}(R)}, \quad (4.40)$$

siendo $R = \{(p_1, \dots, p_N) : 1/M = p_1 \leq p_2 \leq \dots \leq p_N \leq 1\}$ la región factible y Vol la función volumen.

El siguiente teorema muestra la similitud entre ambos problemas, cuando la calidad tiende al óptimo.

Teorema 4.3.11 *Sea $N > 2$ un entero positivo (que representa el tamaño de buffer). Entonces, $E(r) \xrightarrow{r \rightarrow 0} 0$.*

Prueba. Calculemos primero $Vol(R)$. Sabemos que el mismo está comprendido en el $N-1$ cubo contenido en el hiperplano $p_1 = \frac{1}{M}$ y lados $1 - \frac{1}{M}$. Ahora hagamos uso del teorema de Monte Carlo [58], que nos asegura que el volumen de interés es el producto entre el volumen del $N-1$ cubo y la probabilidad de que un muestreo uniforme de las variables $p_i, i = 2, \dots, N$ queden ordenadas en forma creciente. En la distribución uniforme cualquier ordenamiento es igualmente probable, y hay un total de $(N-1)!$ tales ordenamientos. Luego, la probabilidad de que el vector $p_i, i = 2, \dots, N$ quede ordenado en forma creciente es $\frac{1}{(N-1)!}$. En conclusión, el volumen de la región factible es:

$$Vol(R) = \frac{(1 - \frac{1}{M})^{N-1}}{(N-1)!}. \quad (4.41)$$

Busquemos ahora el menor $r' : R_{eq}(r) < B(r', p^*)$, siendo $B(r', p^*)$ la $N-1$ bola de radio r' y centro p^* , con primera coordenada $p_1 = 1/M$. Esto es equivalente a hallar la máxima distancia entre p^* y un punto del hiperplano de calidad constante, contenido a su vez en la región factible. Por el Lema 4.3.9 $r' = \|p^* - q\| = 1 - \frac{k(r)}{1-k(r)} \frac{N-1}{M}$, donde $k(r)$ es la función inversa de la presentada en el Lema 4.3.6. Una consecuencia directa del Lema 4.3.6 es que $k \xrightarrow{r \rightarrow 0} k_{max}$. Entonces:

$$\lim_{r \rightarrow 0} r'(r) = \lim_{k \rightarrow k_{max}} r' = \lim_{k \rightarrow \frac{M}{M+N-1}} 1 - \frac{k(r)}{1-k(r)} \frac{N-1}{M} = 1 - \frac{M}{N-1} \frac{N-1}{M} = 0 \quad (4.42)$$

Por otra parte, el volumen de una $N-1$ bola se expresa en términos de la función gamma $\Gamma(z) = \int_0^{+\infty} t^{z-1} \exp^{-t} dt$ de la siguiente manera:

$$Vol(B(r', p^*)) = \frac{\pi^{\frac{N-1}{2}} r'^{N-1}}{\Gamma(\frac{N+1}{2})} \quad (4.43)$$

Sabemos por la monotonía de la medida volumen, que $Vol(R_{eq}(r)) < Vol(B(r', p^*))$. Entonces, sustituyendo en la definición del error de equivocación se obtiene que:

$$E(r) \leq \frac{Vol(B(r', p^*))}{Vol(R)} = \frac{(N-1)! \pi^{\frac{N-1}{2}} r'^{N-1}}{(1 - \frac{1}{M})^{N-1} \Gamma(\frac{N+1}{2})} \xrightarrow{r \rightarrow 0} 0. \quad (4.44)$$

Hemos encontrado una cota superior de la relación de equivocación que tiende a cero con el radio de calidad r . Como el error de equivocación es no negativo, también tenemos que $E(r) \xrightarrow{r \rightarrow 0} 0$, obteniendo el resultado deseado.

QED

Este último resultado establece que asumiendo una distribución uniforme de las permutaciones sobre la región factible, la probabilidad de encontrar un par de estrategias factibles distintas p_1 y p_2 que logran los óptimos para los problemas 1 y 2 respectivamente, tiende a cero. Esta propiedad explica el hecho de que varias simulaciones realizadas obtuvieron el mismo óptimo en ambos problemas simultáneamente. Obsérvese que la obtención de soluciones idénticas en los dos problemas de optimización en una búsqueda, es un indicador de buena calidad de estas soluciones. En el Cuadro 4.3, se muestran los resultados para un escenario particular de $N = 30, 35, \text{ y } 40$, donde c es la continuidad de Rarest First y el número de pares es $M = 5000$.

	CONTINUIDAD			LATENCIA		
	N=30	N=35	N=40	N=30	N=35	N=40
Rarest First	0.944	0.956	0.964	15.81	20.55	25.39
Greedy	0.720	0.770	0.806	2.30	2.65	2.95
<i>Problema1</i>	0.953	0.965	0.970	5.85	5.78	6.30
<i>Problema2</i>	0.953	0.965	0.970	5.85	5.78	6.30

Cuadro 4.3: Desempeño de distintas estrategias.

Se puede apreciar en el Cuadro 4.3 que los óptimos de los Problemas 1 y 2 logran un mejor desempeño con respecto a las estrategias clásicas. Este hecho justifica la consideración de distinto orden en una consulta.

Con este enfoque hay importantes posibilidades de mejorar el desempeño, aunque resulte prohibitivo una búsqueda exhaustiva para altos valores de tamaño de buffer. En la Sección 4.4 se revisan conceptos básicos de metaheurísticas de importancia para las siguientes secciones. Se formaliza un Problema de Optimización Combinatoria (COP) que captura la calidad de una estrategia de selección de piezas. En la Sección 4.5 se proponen dos nuevos algoritmos para su resolución. El primero se basa en la definición de una métrica sobre el espacio de permutaciones para aplicar una búsqueda local. Aún requiere un diseño previo del vector p de entrada (solamente una vez). En la Sección 4.6 el COP planteado se resuelve eficientemente aplicando la exploración del espacio de soluciones y diversificación con la técnica de Colonia de Hormigas y de Búsqueda Local.

4.4. Formalización del Problema y Metaheurísticas

4.4.1. Introducción

Esta sección formaliza el problema de elección de una estrategia óptima mediante un Problema de Optimización Combinatoria (COP, por sus siglas en inglés). En base al anterior análisis de optimalidad es que se define una función objetivo, y la búsqueda por enumeración es nuevamente factorial. En primera instancia hagamos una revisión de metaheurísticas que se van a aplicar en posteriores algoritmos para su resolución.

4.4.2. Conceptos Básicos

En esta subsección revisaremos algunas definiciones referentes a un Problema de Optimización Combinatoria (COP). Veremos una introducción a las dos metaheurísticas aquí utilizadas, y la metaheurística golosa. El lector puede encontrar en [21] una comparación conceptual de distintas metaheurísticas. Un COP es un problema de maximización (o minimización) de una función sobre un dominio discreto y con restricciones. Más formalmente:

Definición 4.4.1 Sea \mathcal{D} es un espacio de soluciones discreto, $f : \mathcal{D} \rightarrow R^+$ una función y $\{r_1, \dots, r_n\}$ un conjunto de restricciones que deben cumplir las variables de \mathcal{D} , que determinan una región factible S . Un COP (S, f) consiste en determinar el máximo global (o bien el mínimo global) de la función f en el espacio de soluciones factibles S .

La comprensión de la naturaleza del COP es determinante a efectos de diseñar un plan de algorítmica. Ante COPs de alta complejidad pocas veces es posible hallar una solución analítica o algoritmo polinomial que determine la solución exacta. Es frecuente entonces el diseño de algoritmos en base a la aproximación, buscando comúnmente un factor de garantía de proximidad de la solución hallada respecto de la óptima. Otro enfoque muy común y no excluyente con el primero es la aplicación de metaheurísticas. La palabra metaheurística proviene etimológicamente del griego: heurística viene de “heuriskein”, que significa encontrar, y el prefijo “meta” significa en un nivel más alto. Básicamente, una metaheurística es una herramienta de búsqueda aplicable a cualquier problema. La intensificación y diversificación son dos propiedades inherentes a toda metaheurística. La diversificación consiste en lograr un conocimiento del espacio de soluciones mediante la visita a las mismas, o aprendizaje de su estructura. Básicamente es la experiencia del proceso obtenido de recorrer distintas soluciones del espacio. La manera en que se explota este conocimiento entra en la etapa de intensificación. Esta consiste en concebir un subconjunto del subespacio destacado o propiedades a partir del proceso de diversificación, e intensificar la búsqueda sobre este subconjunto o explotar las propiedades del espacio de soluciones.

Las técnicas que constituyen algoritmos con metaheurísticas varían en complejidad desde una simple búsqueda local a un complejo mecanismo de aprendizaje. A continuación presentaremos tres metaheurísticas denominadas Golosa o Greedy, Búsqueda Local y Optimización por Colonia de Hormigas (o ACO por sus siglas en inglés). Vale destacar que existe una gran cantidad de metaheurísticas no abordadas aquí; las tres se eligen por ser conceptualmente valiosas para la comprensión del desarrollo de esta tesis.

4.4.2.1. Greedy

Una de las metaheurísticas más simples e intuitivas es la metaheurística golosa. En la misma se realiza en cada paso una sustitución de la solución que se dispone por aquella cercana con el menor costo posible, procurando así minimizar el costo de la solución global. Una manera natural de tratar un nuevo problema de optimización es comprender el concepto goloso intrínseco en su estructura, y analizar su performance. Su desempeño varía en función de la naturaleza del problema, y normalmente esta metaheurística no logra el óptimo global por sí sola. Es interesante el desempeño de la metaheurística golosa en el TSP, que consiste en hallar

el tour de costo mínimo que visite a todos los nodos de un grafo. En este problema el goloso crea el ciclo tomando en cada oportunidad el enlace más barato, sin crear ciclos. Aquí se puede crear instancias en las que el goloso esté tan lejos como se desee del óptimo.

Otro ejemplo es el de hallar el árbol recubridor de costo mínimo en un grafo conexo con pesos en sus aristas. Aquí la metaheurística golosa es el algoritmo de Kruskal, y logra la solución exacta en tiempo polinomial.

4.4.2.2. Búsqueda Local

La metaheurística de búsqueda local se caracteriza por el requisito de encontrar inicialmente una solución factible al problema de optimización, y comprender su estructura para dar un concepto de estructura de vecindad de esta solución. Básicamente es una noción de proximidad con una solución dada.

Definición 4.4.2 Estructura de Vecindad

Sea Ω un conjunto de soluciones factibles de un problema de optimización combinatoria $\max_{S \in \Omega} \{f(S)\}$. Una estructura de vecindad es una colección $\{\mathcal{N}(S)\}_{S \in \Omega}$ de subconjuntos $\mathcal{N}(S) \subseteq \Omega$ tales que:

- 1) $\forall S, T \in \Omega, \exists S = S_0, \dots, S_k = T : S_i \in \mathcal{N}(S_{i-1}) \quad \forall i = 1, \dots, k$
- 2) Para cada $S \in \Omega$ es posible decidir en tiempo polinomial si existe $T \in \mathcal{N}(S) : f(T) > f(S)$, y hallar esa mejor solución factible, si existe.

Se define análogamente para el caso de un problema de minimización. El Algoritmo 3 muestra la aplicación de una búsqueda local genérica a un COP.

Algoritmo 3 Metaheurística de Búsqueda Local

- 1: $s \leftarrow \text{GenerarSolucionInicial}$
 - 2: **Mientras** Existe mejora posible
 - 3: $s \leftarrow \text{Mejorar}(s, \mathcal{N}(s))$
 - 4: **Fin Mientras**
 - 5: **Retornar** s
-

Nuevamente, la performance de la búsqueda local es muy variable en función de la naturaleza del problema particular. En el caso de una función objetivo convexa en dominio real la búsqueda local se interpreta naturalmente mediante un descenso de gradiente, o variantes según el concepto de vecindad considerado. La búsqueda local permite aproximarse tanto como uno desee al mínimo global mediante esta técnica para una función objetivo convexa. La irregularidad de la función objetivo o presencia de múltiples mínimos locales no globales amenazan a finalizar con una solución subóptima, cuya calidad puede ser muy inferior que la del mínimo global.

Otro ejemplo de problema de distinta naturaleza es de corte máximo o MAXCUT: dado un grafo con costos en sus enlaces, se desea hallar una bipartición de sus nodos en la que la suma de costos entre los enlaces que conectan a esta bipartición sea la máxima posible. Este problema es

también NP -Difícil. Se puede considerar una solución vecina a otra a partir del movimiento de un nodo de un conjunto de la partición a otro. Bajo esta estructura de vecindad, la búsqueda local logra siempre al menos la mitad del óptimo, independientemente de la instancia [86]. En la mayoría de los casos, la semilla o solución inicial juega un rol importante en la búsqueda. Por otra parte, la decisión de la estructura de vecindad y el criterio de parada son cruciales para el éxito, como también para plantear apropiadamente el compromiso entre el tiempo computacional y la calidad de la solución final.

4.4.2.3. Optimización por Colonia de Hormigas (ACO)

ACO es una metaheurística inspirada en la naturaleza y basada en el aprendizaje cooperativo de poblaciones (hormigas artificiales). Las hormigas han llegado al mundo cientos de millones atrás, se estima que hay 10^{16} de ellas [51]. Es sin duda una de las especies más exitosas sobre el planeta. Su organización social altamente estructurada de este insecto es un elemento muy atendido por científicos, y existe una extensa literatura sobre su modo de vida. Una rama de trabajo muy difundida y de actual interés en la ciencia de la computación es la estigmergia (stigmergy en inglés), que es una manera particular de comunicación indirecta utilizada por los insectos para coordinar sus actividades [39]. El estudio de las colonias de hormigas provee modelos de organización distribuida, que son frecuentemente útiles para resolver problemas de optimización y control.

En particular, ACO es una metaheurística aplicada a la optimización que incorpora elementos de aprendizaje para identificar regiones de alta calidad en el espacio de soluciones, aprovechando la estructura distribuida de las hormigas. La esencia se halla en la definición y actualización de una función de feromonas, pues las hormigas tomarán con mayor probabilidad la dirección que posee mayor magnitud de la misma. Fue introducida en 1992 por Marco Dorigo [37]. Procura simular el comportamiento de las hormigas en su hábitat natural para hallar el camino más corto entre su hormiguero y el alimento. La hormiga a pesar de ser un insecto casi ciego, logra encontrar el camino más corto entre el hormiguero y su fuente de alimento y regresar [14]. La comunicación entre hormigas en base a feromonas, que dejan un rastro que sirve de referencia a otras hormigas. Se construye un grafo en el que los nodos representan soluciones factibles y los enlaces sus vínculos. Entonces cada hormiga realiza una exploración del espacio de soluciones recorriendo un camino (subconjunto conectado de soluciones del espacio) y depositando feromonas $\tau_{(i,j)}$ por las conexiones (i, j) y los nodos $\tau_n, n \in V$. Estas feromonas representan una medida de deseo del uso de ese enlace y sesgan la distribución de probabilidades a utilizar por las siguientes hormigas. La actualización de la función de feromonas se realiza en base a la calidad del trayecto realizado por la hormiga, como también un parámetro de heurística η . El Algoritmo 4 muestra una forma genérica de la aplicación de ACO.

Si bien las primeras aplicaciones de esta técnica fueron realizadas sobre TSP, ha logrado mejores desempeños sobre ATSP (TSP asimétrico [66]). La aplicación de ACO sobre ATSP realizada en [40] muestra que la calidad en sus soluciones supera ampliamente a su predecesores. En particular, la performance de ACO fue comparada favorablemente ante el ganador del

Algoritmo 4 Construcción de Permutación de Aproximación

-
- 1: *InicializarFeromonas()*
 - 2: **Mientras** No se cumpla una condición de parada
 - 3: **Para** hormigas = 1 a n
 - 4: $s_a \leftarrow \text{ConstruirSolucion}(\tau, \eta)$
 - 5: **Fin Para**
 - 6: *ActualizarFeromonas()*
 - 7: **Fin Mientras**
-

primer concurso internacional sobre Computación Evolutiva en problemas de ATSP [16].

4.4.3. Problema de Optimización Combinatoria

En base a la Sección 4.3, sabemos que es deseable que la extensión de una consulta sea lo mayor posible.

Definición 4.4.3 La calidad de una permutación dada π es el valor esperado de la extensión de una consulta $Q_\pi = E(X_\pi)$.

Definición 4.4.4 El siguiente Problema de Optimización Combinatoria tiene una permutación como variable de decisión, y procura maximizar la extensión esperada de una consulta con la permutación a elegir, sujeta a las restricciones del problema y definición de estrategia de selección. En términos matemáticos:

$$\max_{\pi} E(X_{\pi}) \quad (4.45)$$

s.a.

$$p_1 = \frac{1}{M} \quad (4.46)$$

$$s_{\pi(1)} = 1 - \frac{1}{M} \quad (4.47)$$

$$p_{i+1} = p_i + (1 - p_i)p_i s_i \quad (4.48)$$

$$s_{\pi(i+1)} = s_{\pi(i)} + (1 - p_{\pi(i)})^2 \quad (4.49)$$

$$S = \{1, \dots, N - 1\}$$

$$\pi : S \rightarrow S, \pi(i) \neq \pi(j), \forall i \neq j$$

Este problema se aborda entonces con metaheurísticas, logrando un compromiso entre el esfuerzo computacional y la calidad de los resultados. En la siguiente sección se incorpora una métrica sobre el espacio de permutaciones, y una estructura de vecindad. Estos elementos permiten realizar una búsqueda local, que retorna mínimos locales del COP recién expuesto. En la Sección 4.6 se explota que la naturaleza de búsqueda de solución es análoga a aquella selección de la ruta óptima en el conocido problema TSP (Travelling Salesman Problem o

Problema del Vendedor Ambulante) [46, 59]. Se traduce el COP en un TSP asimétrico o ATSP. Se resuelve este último mediante una metaheurística más elaborada, basada en Optimización por Colonia de Hormigas (ACO) y una fase de Búsqueda Local. El tour correspondiente va a ser la elección de la permutación en el COP, problema central de este capítulo. Con unas simples comparaciones finales con las estrategias clásicas y Mixta, se puede apreciar el potencial de esta nueva traducción del problema, como también la eficacia de la metaheurística aplicada.

4.5. Algoritmo de Búsqueda Local

En esta sección se introducen elementos para enfrentar el COP propuesto en la Def. 4.4.4. El espacio de soluciones se enriquece con una distancia y una noción de estructura de vecindad, que permite engendrar un algoritmo basado en Búsqueda Local.

4.5.1. Espacio Métrico con estructura de vecindad

Definición 4.5.1 La distancia $d(\pi, \pi^*)$ entre dos permutaciones π y π^* es el mínimo número de intercambios de elementos necesarios para transformar π en π^* .

Daremos a continuación dos propiedades básicas de la distancia entre permutaciones, que permiten descubrir el potencial de su aplicación para definir una posterior búsqueda local en el espacio de permutaciones.

Proposición 4.5.2 Llamemos Π al espacio de permutaciones. Luego (Π, d) es un espacio métrico, es decir que cumple con las siguientes tres afirmaciones:

- 1) $\forall \pi, \pi^* \in \Pi, d(\pi, \pi^*) \geq 0$; $d(\pi, \pi^*) = 0 \leftrightarrow \pi = \pi^*$ (no-negatividad y reflexividad nula)
- 2) $\forall \pi, \pi^* \in \Pi, d(\pi, \pi^*) = d(\pi^*, \pi)$ (simetría)
- 3) $\forall \pi_a, \pi_b, \pi_c \in \Pi, d(\pi_a, \pi_b) + d(\pi_b, \pi_c) \geq d(\pi_a, \pi_c)$ (desigualdad triangular)

Prueba. La primera afirmación es evidente: la función $d(\pi, \pi^*)$ por definición toma valores naturales. Si $\pi = \pi^*$ es posible llevar π a π^* con cero intercambios, luego $d(\pi, \pi^*) = 0$. Recíprocamente, si $d(\pi, \pi^*) = 0$ con cero intercambios logramos llevar π a π^* , luego π permanece incambiada para obtener a π^* , y en consecuencia son la misma permutación.

Para probar ahora la simetría consideremos dos permutaciones π y π^* arbitrarias. Denotemos $d = d(\pi, \pi^*)$. Por definición existe una secuencia de permutaciones de largo d : $\{\pi_0, \pi_1, \dots, \pi_d\}$ tal que $\pi_0 = \pi$, $\pi_d = \pi^*$ y $d(\pi_i, \pi_{i+1}) = 1$, $\forall i = 1, \dots, d - 1$. Además, si $d(\pi_i, \pi_{i+1}) = 1$ alcanza con un intercambio para llevar π_i a π_{i+1} , y este intercambio también vale para llevar de π_{i+1} a π_i . Esto permite afirmar que la secuencia inversa $\{\pi_d, \pi_{d-1}, \dots, \pi_0\}$ lleva de $\pi_d = \pi^*$ a $\pi_0 = \pi$. Con este razonamiento se tiene que cualquier secuencia inversa de π a π^* lleva de π^* a π . En particular, el largo de la secuencia mínima de π a π^* coincide con el largo de la secuencia mínima de π^* a π . Esto prueba que $d(\pi, \pi^*) = d(\pi^*, \pi)$.

Para probar la desigualdad triangular, consideremos tres permutaciones arbitrarias π_a, π_b y π_c . Si denotamos $l_1 = d(\pi_a, \pi_b)$ y $l_2 = d(\pi_b, \pi_c)$ por definición de la distancia sabemos entonces

que existen dos secuencias de permutaciones $\{\pi_0, \pi_1, \dots, \pi_{l_1}\}$ y $\{\pi_0^*, \pi_1^*, \pi_2^*, \dots, \pi_{l_2}^*\}$ tales que $\pi_0 = \pi_a, \pi_{l_1} = \pi_b, \pi_0^* = \pi_b, \pi_{l_2}^* = \pi_c$ y además $d(\pi_i, \pi_{i+1}) = 1, \quad \forall i = 1, \dots, l_1 - 1$ y $d(\pi_i^*, \pi_{i+1}^*) = 1, \quad \forall i = 1, \dots, l_2 - 1$. Sabemos por la definición de $d(\pi, \pi^*)$ además que estas secuencias son del largo el mínimo posible. Ahora podemos construir la siguiente secuencia de permutaciones concatenación de las anteriores: $\{\pi_a, \pi_1, \dots, \pi_{l_1-1}, \pi_b, \pi_1^*, \pi_2^*, \dots, \pi_{l_2-1}^*, \pi_c\}$. Esta concatenación permite asegurar que la distancia entre las permutaciones π_a y π_c es no mayor que el largo de esta secuencia, es decir: $l_1 + l_2$. Obtenemos así que $d(\pi_a, \pi_b) + d(\pi_b, \pi_c) \geq d(\pi_a, \pi_c)$. Luego la desigualdad triangular se satisface para cualquier terna de permutaciones, y estamos en condiciones de afirmar que d es una métrica, o equivalentemente (Π, d) es un espacio métrico, como se quería demostrar.

QED

Proposición 4.5.3 Consideremos el problema de maximizar $E(X_\pi)$. Dada una permutación π (solución factible) definamos $\mathcal{N}(\pi) = \{\pi^* : d(\pi, \pi^*) = 1\}$ como el conjunto vecindad de la permutación π . Luego $\{\mathcal{N}(\pi)\}$ es una estructura de vecindades para el problema de maximización $\max_\pi E(X_\pi)$

Prueba. Consideremos dos permutaciones arbitrarias π y π^* . Si d es la cantidad de intercambios necesario para llevar π en π^* , entonces existe una secuencia de permutaciones $\{\pi_0, \pi_1, \dots, \pi_d\}$ tales que $\pi_0 = \pi, \pi_d = \pi^*$ y $d(\pi_i, \pi_{i+1}) = 1, \quad \forall i = 1, \dots, d - 1$. Por la definición de vecindad tenemos que $\pi_{i+1} \in \mathcal{N}(\pi_i), \quad \forall i \in \{0, \dots, d - 1\}$. Esto prueba la condición (1) de la definición de estructura de vecindades. Falta probar que es posible decidir en tiempo polinomial si existe una permutación de la vecindad de una dada que es mejor. Primero que nada observemos que la cantidad de permutaciones vecinas a una dada coincide con todos los intercambios de dos índices: $C_2^{N-1} = \frac{(N-1)(N-2)}{2}$ que es de orden cuadrático en el tamaño de las permutaciones. Alcanza para demostrar la propiedad (2) de las estructuras de vecindad que exista un algoritmo polinomial que permita estimar a $E(X_\pi)$ tanto como se desee (y así poder decidir qué permutaciones son mejores). El cálculo de $E(X_\pi)$ dada la permutación π se puede realizar resolviendo el Sistema $S(\pi)$ (Def. 4.2.4). Este sistema no lineal se puede resolver mediante el método de Newton-Raphson, logrando un esquema de aproximación polinomial a su solución ². Puesto que para cualquier $\epsilon > 0$ podemos estimar la calidad de una permutación, si existe $\pi^* \in (N)(\pi) : E(X_{\pi^*}) > E(X_\pi)$, tomando $\epsilon < \frac{E(X_{\pi^*}) - E(X_\pi)}{2}$ podemos estimar en tiempo polinomial con error menor que ϵ tanto a $E(X_{\pi^*})$ como a $E(X_\pi)$ y decidir correctamente que $E(X_{\pi^*}) > E(X_\pi)$. Por último nótese que siempre que exista tal permutación mejor en una vecindad es posible encontrarla en tiempo polinomial, pues basta con revisar todas las $\frac{(N-1)(N-2)}{2}$ vecinas y determinar en cada una si es mejor que la primera.

QED

²El dominio de posibles valores del vector (p_2, \dots, p_N) es un $N - 1$ cubo, que es compacto y convexo. Bajo estas condiciones, si se elige adecuadamente el punto inicial de la solución y existe una contracción (válido si existe una norma menor que 1 para la matriz jacobiana, no singular), entonces el método de Newton-Raphson para sistemas converge con orden al menos cuadrático. Aún no se ha probado que el método de Newton-Raphson genere una contracción, a pesar que en todas las resoluciones del sistema no lineal hayan resultado convergentes partiendo del vector nulo.

4.5.2. Algoritmo de Búsqueda Local

Haciendo uso de las propiedades de la familia de permutaciones y la estructura de vecindades que nos brinda la distancia entre permutaciones, es posible adaptar el Algoritmo 2 a otro que agrega una búsqueda local y se orienta ya a maximizar la función objetivo del COP aquí propuesto. El Algoritmo 5 acepta una probabilidad de ocupación deseada y construye en una primera etapa de “aproximación” una permutación la cual se espera que logre una probabilidad de ocupación p^* próxima a aquella de la entrada. Finalmente, una Búsqueda Local es aplicada, que sustituye en cada etapa la permutación presente por la mejor permutación vecina. Se explota así el potencial de la búsqueda local, retornando una permutación que es óptima localmente bajo la estructura de vecindades generadas por las permutaciones a distancia 1 en el espacio métrico (Π, d) . Se observa que si bien eliminamos la necesidad de redefinir en varias etapas el vector de probabilidades a la entrada (como ocurre en el Algoritmo 2), es necesario realizar el diseño de este vector de entrada p una vez. Luego la búsqueda local evita la recurrencia, retornando ya una permutación a la salida.

Algoritmo 5 Recibe una probabilidad de ocupación. Aplica la Propiedad de Aproximación de Estrategias para hallar en primera instancia una permutación intermedia π . Por último se realiza una Búsqueda Local para mejorar la permutación antes obtenida.

Entradas:

Probabilidad de ocupación: $p_i, i = 1, \dots, N$

Número de iteraciones: n

Salida:

Permutación: π

- 1: **Para** $i = 1$ a $N - 1$
 - 2: $s_{ideal}(i) = \frac{p_{i+1} - p_i}{(1 - p_i)p_i}, i = 1, \dots, N - 1$ // Estrategia que logra p .
 - 3: **Fin Para**
 - 4: $\pi \leftarrow Algoritmo1(s_{ideal})$
 - 5: $(s, p^*) \leftarrow ResolverSistemaS(\pi)$
 - 6: $E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} i(p_{\pi(i)+1}^* - p_{\pi(i)}^*)$
 - 7: // Fase de Búsqueda Local
 - 8: **Para** $i = 1$ a n
 - 9: $\pi_{aux} \leftarrow MejorVecina(\pi)$
 - 10: **Si** $E(X_{\pi_{aux}}) > E(X_\pi)$ **entonces**
 - 11: $\pi \leftarrow \pi_{aux}$
 - 12: **Fin Si**
 - 13: **Fin Para**
 - 14: **Retornar** π
-

A efectos comparativos, ingresemos la misma probabilidad de ocupación que en el Algoritmo 2. Esto es con $M = 1000, N = 40$ la probabilidad de ocupación p . La Figura 4.11 comprueba que la medida de optimalidad $E(X_\pi)$ da prioridad absoluta a la continuidad, teniendo la salida del Algoritmo 5 una continuidad prácticamente de 1. No obstante, su latencia es muy amplia aún.

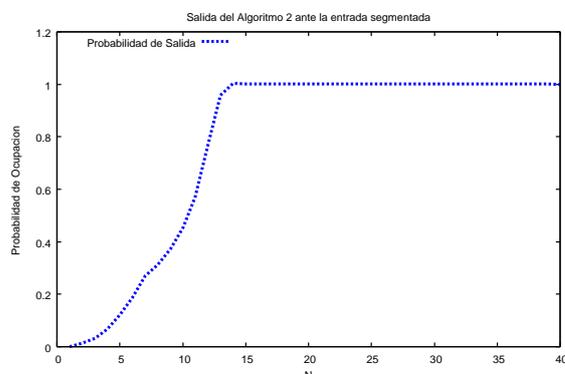


Figura 4.11: Salida del Algoritmo 2. Se observa la prioridad dada a la continuidad, penalizando la latencia.

4.6. Búsqueda de Permutación basada en ACO

4.6.1. Introducción

En esta sección veremos una manera de construir y obtener un problema de ATSP (TSP asimétrico) que capture el COP propuesto en la Sección 4.4. La novedad se halla no solamente en el significado de este grafo, sino en el proceso de construcción del mismo, que difiere del clásico realizado en los sistemas de colonias de hormigas. Un grupo de hormigas artificiales van a cooperar para construir este grafo, aprovechando la experiencia que ya disponemos del problema. Una vez construido este grafo, un segundo grupo de hormigas rastrean ciclos de alta calidad, de modo similar al aplicado en ACO. Este ciclo se puede interpretar como una solución del COP, mediante una biyección ciclo-permutación. Por último se aplica una etapa de búsqueda local al ciclo resultante, obteniendo soluciones de mayor calidad que en la propuesta de resolución anterior, que utilizaba únicamente una búsqueda local.

Esta sección se organiza de la siguiente manera. La Subsección 4.6.2 muestra brevemente que es posible traducir un una permutación en un ciclo, y recíprocamente. A continuación se introduce el Algoritmo Principal, que aplica ACO para hallar ciclos de mejor calidad. Se brinda un pseudocódigo de las funciones que se utilizan en este algoritmo, detallando a su vez la implementación de ACO. Por último en la Subsección 4.6.3 se estudia el desempeño del Algoritmo Principal, incluyendo un análisis del esfuerzo computacional realizado y una comparación con las estrategias Greedy, Rarest First y Mixta.

4.6.2. Biyección entre una permutación y un ciclo

En primera instancia veremos una biyección plausible de ser utilizada para traducir el problema de búsqueda de permutaciones en el espacio métrico (Π, d) , a la búsqueda del tour de costo mínimo en un N -clique con costos asimétricos en sus enlaces.

Proposición 4.6.1 *Un N -clique permite obtener una biyección entre un ciclo dirigido que visita a todos sus nodos y una permutación de $\{1, \dots, N - 1\}$.*

Prueba. Sean K_N un N -clique, con nodos etiquetados $\{1, \dots, N\}$, y N el nodo auxiliar, donde inician todos los ciclos dirigidos. La igualdad de cardinalidad entre una permutación sobre el conjunto de nodos $\{v_1, \dots, v_{N-1}\}$ y el ciclo se halla en el orden de los nodos visitados, sin considerar el nodo v_N . En efecto, tomemos un ciclo con sentido que visita a todos los nodos e inicia en el auxiliar v_N : $C = \{v_N, v_1, v_2, \dots, v_{N-1}, v_N\}$, donde los v_i representan distintos nodos del clique. Entonces $\pi(i) = v_i, \forall i = 1, \dots, N-1$ es biyectiva. Recíprocamente, a cada permutación $\pi(i)$ le corresponde un único ciclo con sentido $C = \{v_N, v_{\pi(1)}, \dots, v_{\pi(N-1)}, v_N\}$. La existencia de estos ciclos en el grafo está asegurada pues en el clique todos sus nodos están conectados directamente.

QED

Obsérvese que a efectos de traducir nuestro COP, el TSP debe ser asimétrico (ATSP); en caso contrario Greedy y Rarest First definirían el mismo ciclo, pero son estrategias de consulta diferentes.

4.6.3. El Algoritmo Principal

Para poder asignar distancias a las aristas, hagamos uso de un mecanismo de aprendizaje basado en la exploración de hormigas, aproximando el enfoque a aquél de Optimización por Colonia de Hormigas (ACO). Enumeremos los nodos del clique $1, \dots, N$, y coloquemos una hormiga en el nodo auxiliar N (denotaremos indistintamente al nodo y a su respectivo índice). Llevaremos el Problema de Optimización Combinatoria a una instancia del bien conocido modelo TSP en su modalidad asimétrica, tal como se esquematiza en el Algoritmo 6 que se detalla a continuación. El primer bloque consiste en las líneas 1 a 3, y tiene la finalidad de construir el grafo completo, aprovechando la biyección indicada en la Proposición 4.6.1 para traducir el COP propuesto en la búsqueda del tour de largo mínimo. En primera instancia (línea 2) se inicializa los largos de todas las aristas mediante la función *InicializarAristas(hormigas)*. Esta inicialización hace uso de la cooperación de hormigas artificiales, donde cada una realiza un tour sesgado y las distancias entre las aristas se define en función de la calidad $E(X_\pi)$ de cada ciclo obtenido (se explica esta función en detalle más adelante). A continuación en la línea 3 se hace la inicialización de las feromonas de los enlaces del grafo. Vamos a poder apreciar en el posterior estudio de la función *InicializarFeromonas(SubFamilia)* (que se detalla en 8) que esta inicialización aprovecha nuestra experiencia previa en el problema de búsqueda de permutaciones. Se va a utilizar la subfamilia de permutaciones que presenta un pico para dejar rastros de feromonas en los enlaces visitados en estos ciclos, según la calidad $E(X_\pi)$ de los mismos.

El siguiente bloque comprende desde la línea 4 a la 7. Refiere a la aplicación de ACO para obtener una permutación a partir del grafo asimétrico antes construido y el cálculo posterior de la calidad de la permutación π salida de la función *AplicarACO* de la Línea 5 (que se detalla más adelante). La Línea 6 aplica el método de Newton-Raphson para resolver el sistema no lineal y obtener la probabilidad de ocupación p^* y la estrategia s asociadas a la permutación π

de entrada. Inmediatamente se calcula la calidad de la permutación π en la Línea 7. Las hormigas encuentran el mejor camino a través de feromonas. El último bloque comprende las líneas 8 a 15, y consiste en la etapa de Búsqueda Local. Aprovecha el potencial que (Π, d) es un espacio métrico, como también que esta métrica define una estructura de vecindades. La permutación producida mediante la aplicación de la función *ACO* es sustituida en cada iteración por la mejor de sus vecinas. Este proceso termina cuando ya no es posible mejorar más, habiendo hallado una permutación óptima local, o bien si se alcanza un número máximo de iteraciones. Estudiaremos a continuación en detalle a las Funciones *InicializarAristas(hormigas)*, *InicializarFeromonas(SubFamilia)* y *AplicarACO* en este mismo orden.

Algoritmo Principal 6 El primer bloque traduce el problema a un ATSP. Luego se aplica ACO para finalmente retornar una permutación óptima local mediante una fase de búsqueda local.

Entradas:

Subfamilia de permutaciones: *SubFamilia*

Número de hormigas: *hormigas*

Número de iteraciones en ACO: *iteraciones*

Número de iteraciones máximo en Búsqueda Local: *n*

Salida:

Permutación π

```

1: // Construcción de la red
2: Distancias = InicializarAristas(hormigas)
3: Feromonas = InicializarFeromona(SubFamilia)
4: // Resolución del ATSP resultante
5:  $\pi$  = AplicarACO(Distancias, Feromonas, iteraciones)
6:  $(s, p^*) \leftarrow$  ResolverSistemaS( $\pi$ )
7:  $E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} i(p_{\pi(i)+1}^* - p_{\pi(i)}^*)$ 
8: // Fase de Búsqueda Local
9: Para  $i = 1$  a  $n$ 
10:    $\pi_{aux} \leftarrow$  MejorVecina( $\pi$ )
11:   Si  $E(X_{\pi_{aux}}) > E(X_\pi)$  entonces
12:      $\pi \leftarrow \pi_{aux}$ 
13:   Fin Si
14: Fin Para
15: Retornar  $\pi$ 

```

4.6.3.1. Función *InicializarAristas*

Conocida la similitud entre un ciclo orientado en un N -clique y una permutación, la primera etapa del Algoritmo Principal consiste en inicializar las distancias de las aristas del clique, hecho realizado mediante la función *InicializarAristas* (ver Función 7). El primer bloque es de inicialización, y está dado por las líneas 1 a 3. Se define al inicio las distancias entre todos los enlaces iguales a 1, donde *Distancias* es una matriz cuadrada de tamaño N que almacena

en la entrada (i, j) la distancia de i a j , en ese sentido (recordar que se espera una matriz final asimétrica). Luego una calidad de referencia (la de Rarest First) inicializa a Q_{max} , que va a ser a lo largo de la función la calidad máxima encontrada hasta el momento. Posteriormente comienza el bloque de exploración de las hormigas de los distintos ciclos, que se interpretan como soluciones a partir de la biyección probada en 4.6.1. El ciclo realizado por cada hormiga comienza siempre por el nodo auxiliar N . Cada hormiga elige probabilísticamente el siguiente nodo a visitar sin hacer ciclos, tal como se indica en la línea 9. La traducción del ciclo en una permutación se aplica tomando en cuenta el orden de visita de los nodos no auxiliares, como se aprecia en la línea 11. Las líneas 12 a 16 simplemente calculan la calidad de la permutación obtenida mediante el ciclo estocástico realizado por la hormiga artificial, y actualizan la calidad máxima Q_{max} , en caso de ser la calidad del nuevo ciclo superior a la anterior. El bloque de líneas 17 a 20 es fundamental para valorar el aprendizaje obtenido por cada una de las hormigas. Inmediatamente luego de cada ciclo obtenido por una hormiga, se actualizan las distancias de todos los enlaces visitados por la misma. La distancia de cada uno de estos enlaces es inversamente proporcional a la calidad del ciclo obtenido: de esta manera se sesga a visitar los enlaces con más calidad (más cortos) por la siguiente hormiga, y las restantes. Obsérvese también de la actualización en la línea el factor $10(N - j)$, que brinda mayor peso a los enlaces visitados al principio del ciclo. De esta manera, se expande la brecha entre ciclos de mejor calidad de otros de menor calidad, invitando a las siguientes hormigas a elegir el camino con mayor calidad desde el primer paso.

Antes de pasar a analizar la siguiente función del Algoritmo Principal, contemos el mínimo número de hormigas que se necesitan para inicializar correctamente las distancias entre todos los enlaces. Se va a realizar tantos ciclos como cantidad de hormigas, y el N -clique posee $C_2^N = \frac{N(N-1)}{2}$ enlaces. Cada hormiga visita un total de $N - 1$ enlaces (el último enlace (x_{N-1}, N) del ciclo con sentido no se actualiza). Luego se requiere que el número de hormigas supere al menos $\frac{N}{2}$. Debido al sesgo de los enlaces y visita frecuente con repeticiones, es normal tomar un número de hormigas superior a $2N$.

4.6.3.2. Función *InicializarFeromonas*

Una vez que se dispone del clique con distancias asimétricas definidas entre todo par de nodos i y j , se procede asignando feromonas a cada enlace. Esto tiene la finalidad de preparar el grafo para la posterior aplicación de ACO, en el que las hormigas rastrean ciclos de mayor calidad en base a las mismas. A efectos de inicializar la feromona de cada enlace, se elige como entrada la subfamilia de permutaciones que posee un pico, debido a la experiencia de diseño de las secciones anteriores que nos muestra su riqueza y variada calidad. El bloque de inicialización consiste en las líneas 1 a 3. Simplemente asigna feromona con valor 1 a todos los enlaces del clique (*Feromona* es una matriz cuadrada de tamaño N , y su entrada (i, j) representa la feromona del enlace con sentido $e = (i, j)$). A continuación el bloque de realización de ciclos por las hormigas y actualización de feromonas es similar al proceso de inicialización de distancias. Las diferencias se hallan en que en este caso se explota el concimiento de ciclos destacados, y entonces las hormigas recorren estos ciclos determinísticamente. Además, luego de cada ciclo se actualiza la función de feromona de cada enlace visitado en proporción con

Función *InicializarAristas* 7 Inicialización de las distancias entre los nodos de la red. Cada hormiga realiza un ciclo estocástico considerando las probabilidades de cada camino según las distancias del paso anterior.

Entradas:Número de hormigas: *hormigas*Tamaño de buffer: *N***Salida:**Distancias entre nodos: *Distancias*

```

1: // Inicialización
2: Distancias =  $1_{N \times N}$ 
3:  $Q_{max} = E(X_{RarestFirst})$ 
4: // Ciclo de cada hormiga
5:  $x_0 = N$  // Todas las hormigas inician su ciclo en el nodo auxiliar N
6: Para  $i = 1$  a hormigas
7:   Para  $j = 1$  a  $N - 1$ 
8:     // Elegir el siguiente nodo  $x_j$  de la hormiga  $i$  según las siguientes probabilidades:
9:     
$$p_{x_j} = \frac{Distancias(x_{j-1}, x_j)^{-1}}{\sum_{y \text{ Posibles}} Distancias(x_{j-1}, y)^{-1}}$$

10:    Fin Para
11:     $\pi \leftarrow \{x_1, \dots, x_{N-1}\}$  // la permutación ya obtenida por la hormiga  $i$ 
12:     $(s, p) \leftarrow ResolverSistemaS(\pi)$ 
13:     $E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} i(p_{\pi(i)+1}^* - p_{\pi(i)}^*)$ 
14:    Si  $(E(X_\pi) > Q_{max})$  entonces
15:       $Q_{max} = E(X_\pi)$ 
16:    Fin Si
17:    // Actualizar las distancias entre las aristas usadas
18:    Para  $j = 1$  a  $N - 1$ 
19:       $Distancias((x_{j-1}, x_j)) = \frac{10(N-j)Q_{max}}{E(X_\pi)}$ 
20:    Fin Para
21:  Fin Para
22: Retornar Distancias

```

la calidad del ciclo así obtenido. Esto se elige porque los enlaces con mayor cantidad de feromonas son lo que se van a invitar a visitar con mayor probabilidad a las hormigas, durante la aplicación de ACO.

Función *InicializarFeromonas* **8** Las feromonas para la posterior aplicación de ACO se inicializan según la experiencia obtenida de la Subfamilia de permutaciones.

Entradas:

SubFamilia de permutaciones: *SubFamilia(I, J)*

Tamaño de Buffer: N

Salida:

Feromonas de cada arista: *Feromona*

```

1: // Inicialización
2:  $Feromona = 1_{N \times N}$ 
3:  $Q_{max} = Q_{RarestFirst}$ 
4: // Ciclo de cada hormiga
5: Para  $i = 0$  a  $N - 1$ 
6:   Para  $j = 0$  a  $N - i - 1$ 
7:      $\pi = SubFamilia(i, j)$ 
8:      $(s, p^*) \leftarrow ResolverSistemaS(\pi)$ 
9:      $Q = E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} i(p_{\pi(i)+1}^* - p_{\pi(i)}^*)$ 
10:    Si  $(Q > Q_{max})$  entonces
11:       $Q_{max} = Q$ 
12:    Fin Si
13:    // Actualizar las feromonas de las aristas utilizadas
14:    Para  $k = 1$  a  $N - 1$ 
15:       $Feromonas(enlace(k)) = \frac{10(N-k)Q}{Q_{max}}$ 
16:    Fin Para
17:  Fin Para
18: Fin Para
19: Retornar Feromonas

```

4.6.3.3. Implementación de ACO

Una vez inicializados los costos y feromonas de todas las aristas del N -clique (mediante la funciones *InicializarAristas* e *InicializarFeromonas*), se debe resolver el ATSP correspondiente. Puesto que el mecanismo de exploración del problema es similar a aquel utilizado por las hormigas, sin mencionar su bajo esfuerzo computacional y simplicidad de implementación, se ha optado aquí por diseñar una metaheurística basada en ACO, mezclada con una búsqueda local.

Antes de entrar en detalles de la implementación de la función *AplicarACO* vale considerar previamente los clásicos parámetros para la implementación de ACO en TSP, tal como se puede encontrar en la literatura especializada [8, 38, 40, 42]. En todas estas se definen al menos 4

parámetros básicos (además de la función de feromonas y largos de cada enlace ya comentados anteriormente). Uno es el número de hormigas a utilizar, siempre presente en toda implementación de ACO. Dos parámetros denominados α y β son respectivamente los exponentes de prioridad a la feromona y a la distancia. Son variables positivas que permiten dar mayor peso al rastreo mediante feromonas, o bien a caminos más cortos. Más específicamente, si x_j es el nodo actual de la hormiga, esta elige el siguiente nodo x_{j+1} según las siguientes probabilidades [40] (ver la Función *AplicarACO* detallada en 9):

$$p_{x_{j+1}} = \frac{Feromona(x_j, x_{j+1})^\alpha Distancias(x_j, x_{j+1})^{-\beta}}{\sum_{Posibles\ y} Feromona(x_j, y)^\alpha Distancias(x_j, y)^{-\beta}} \quad (4.50)$$

donde α y β son los parámetros de prioridad a las distancias y feromonas respectivamente. Por último, un parámetro usual para la actualización de la función de feromonas luego de cada ciclo es el factor de evaporación ρ . Este factor brinda una medida del impacto en la nueva actualización de la feromona: con $\rho = 1$ el valor de feromona presente del historial de visitas de cada enlace se borra por completo, siendo ya el nuevo que se desprende de la calidad del ciclo obtenido. En cambio, un valor de $\rho = 0$ significa valores estáticos de las feromonas. Más concretamente, si $\tau_{anterior}$ representa el valor de la feromona de un enlace previo a la actualización y $\tau_{presente}$ es la feromona que depende de la calidad del ciclo presente, entonces la actualización de la función de feromonas se realiza a partir de la siguiente ecuación:

$$\tau_{nueva} = (1 - \rho)\tau_{anterior} + \rho\tau_{presente}$$

Ahora dirijamos la atención a la Función 9 *AplicarACO*, en la que se introducen los anteriores conceptos. Los parámetros de entrada son las matrices de feromonas y distancias obtenidas mediante *InicializarAristas* e *InicializarFeromonas* en las etapas anteriores y los parámetros *hormigas*, α , β y ρ antes expuestos. La salida es una permutación π . El bloque de inicialización consiste en las líneas 1, 2 y 3. La línea 2 toma como referencia la calidad de Rarest First normalizada por la latencia. Obsérvese esta normalización, que se justifica por la prioridad absoluta de la continuidad en $E(X_\pi)$; se espera con este artificio obtener soluciones con latencias más reducidas. Se define el punto de partida de todas las hormigas N (nodo auxiliar), en la línea 3. Cada vez que una hormiga finaliza su ciclo se actualiza la función de feromonas, utilizando su calidad basada en $E(X_\pi)$, pero dando prioridad a la latencia. Esto se elige así pues la subfamilia tiene un margen importante para mejorar la latencia, pero no así la continuidad. Los ciclos realizados por las hormigas son estocásticos, y se construyen en las líneas 4 a 16. Cada hormiga avanza un paso aleatoriamente y sin hacer ciclos haciendo uso de la expresión (4.50). A continuación se actualiza la calidad máxima Q_{max} , en caso de haber obtenido un ciclo con mejor calidad normalizada que la mejor hasta el momento. Al final de cada iteración (luego de cada ciclo completado por cada hormiga) se efectúa una actualización de la función de feromonas, en los enlaces visitados (líneas 17 a 20). La expresión de actualización de feromonas para cada una de las aristas visitadas depende de la calidad $Q = E(X_\pi)$ del ciclo obtenido y de la calidad del mejor ciclo obtenido hasta el momento Q_{max} , de la siguiente manera:

$$Feromonas((x_j, x_{j+1})) = (1 - \rho)Feromonas((x_j, x_{j+1})) + \rho \frac{10(N - j)Q}{Q_{max} \sum_{i=1}^N p_i}$$

Se sigue el esquema tradicional de evaporación de feromonas, dada por el parámetro ρ : $0 \leq \rho \leq 1$. Nótese el factor $10(N - k)$. A medida que la hormiga avanza su ciclo, la ponderación de la feromona es menor. Ese proceso de escalerización permite reducir la probabilidad de desvío de caminos de alta calidad, si es que se encuentran en este proceso estocástico. Por último, se construye en forma golosa el mejor ciclo a la salida. Para su construcción se toma en cada oportunidad el enlace más visitado, sin formar ciclos.

Recordemos que en el Algoritmo Principal el ciclo salida de *Aplicar ACO* entra en la Fase de Búsqueda Local, en la cual se sustituye en cada iteración por el mejor de sus ciclos vecinos. Este proceso finaliza cuando ya no es posible mejorar en el paso siguiente (y entonces esta permutación final es mejor que cualquiera de sus vecinas). La permutación salida de la fase de Búsqueda Local es un mínimo local (si el tope de iteraciones no limita), y es la salida final de todo el algoritmo.

4.6.3.4. Resultados y comparaciones

La adaptación de los parámetros se basa en [42], y se ajustan en base a este problema en particular. La implementación final aplica el Algoritmo Principal con $\alpha = 0,4$, $\beta = 1,5$ y $\rho = 0,5$, y 100 hormigas. La Figura 4.12 muestra una comparación entre previas estrategias y la salida del Algoritmo Principal para el caso en que $N = 30$ y $M = 100$. El Cuadro 4.4 muestra que la salida logra una excelente continuidad y al mismo tiempo una latencia comparable con Greedy (última con continuidad pobre).

Cuadro 4.4: Desempeño de distintas estrategias.

Estrategia	Continuidad	Latencia
Rarest First	0.9571	21.0011
Greedy	0.9020	4.1094
Mixta	0.9953	11.1253
Nueva Permutación	0.9998	7.9821

4.6.4. Análisis de la cantidad de operaciones

A efectos de comprender el compromiso entre el esfuerzo computacional y la calidad de soluciones que ofrece el Algoritmo Principal realicemos un análisis de la cantidad de operaciones. Remitimos al lector al Algoritmo 6. Vamos a tomar como unidad de cantidad de operaciones al número de evaluaciones de calidad $E(X_\pi)$, que se corresponde con una resolución del

Función *AplicarACO* **9** Las feromonas de la aplicación de ACO son inicializadas según la experiencia de buenos resultados obtenidos por la SubFamilia de permutaciones.

Entradas:Costo de cada arista: *Distancias*Feromonas de las aristas: *Feromonas*Cantidad de hormigas: *hormigas*Prioridad a las feromonas: α Prioridad a las distancias: β Parámetro de evaporación: ρ **Salida:**Mejor ciclo logrado: π

1: // Inicialización

2: $Q_{max} = \frac{E(X_{RarestFirst})}{LatenciaRarestFirst}$

3: $x_0 = N$ // Todas las hormigas comienzan en el nodo auxiliar N

4: // Ciclo aleatorio de hormigas

5: **Para** $i = 1$ a *hormigas*6: **Para** $j = 1$ a $N - 1$ 7: // Elegir el siguiente nodo x_j de la hormiga i según las siguientes probabilidades:

8:
$$p_{x_j} = \frac{Feromona(x_{j-1}, x_j)^\alpha Distancias(x_{j-1}, x_j)^{-\beta}}{\sum_{y \text{ Posibles}} Feromona(x_{j-1}, y)^\alpha Distancias(x_{j-1}, y)^{-\beta}}$$

9: **Fin Para**10: $\pi \leftarrow \{x_1, \dots, x_{N-1}\}$ // la permutación ya obtenida por la hormiga i 11: $(s, p^*) \leftarrow ResolverSistemaS(\pi)$

12: $Q = \frac{E(X_\pi)}{\sum_{i=1}^N p_i}$

13: **Si** ($Q > Q_{max}$) **entonces**

14: $Q_{max} = Q$

15: **Fin Si**16: **Fin Para**

17: // Actualizar las feromonas de las aristas usadas

18: **Para** $j = 1$ a $N - 1$

19: $Feromonas((x_{j-1}, x_j)) = (1 - \rho)Feromonas((x_{j-1}, x_j)) + \rho \frac{10(N-j)Q}{Q_{max} \sum_{i=1}^N p_i}$

20: **Fin Para**

21: // Hallar el ciclo a la salida

22: $y_0 = N$

23: **Para** $j = 1$ a $N - 1$

24: $y_i = NodoMasVisitadoDesde(y_{i-1})$

25: $\pi(i) = y_i$

26: **Fin Para**27: **Retornar** π

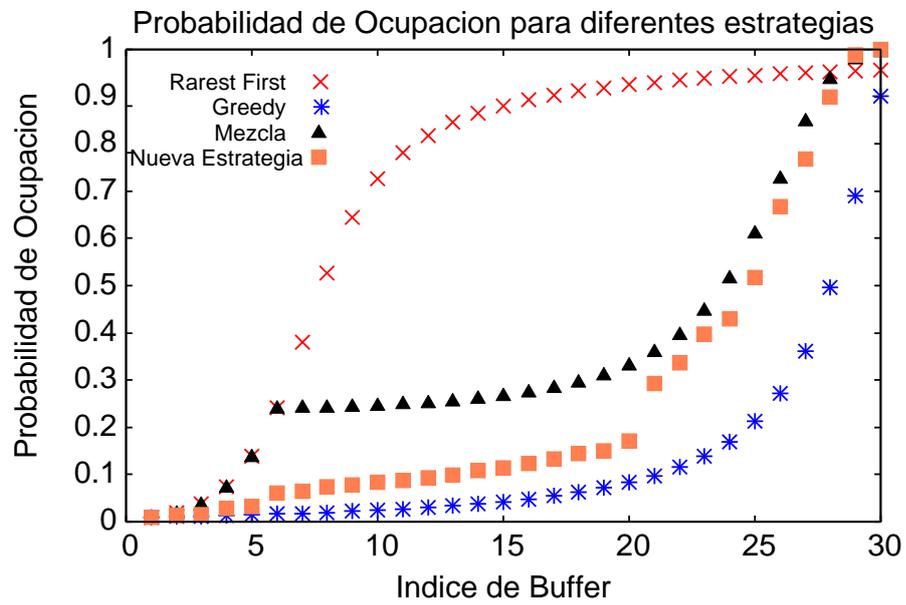


Figura 4.12: Comparación entre distintas estrategias

sistema no lineal $S(\pi)$. El siguiente resultado resume la etapa que impone el mayor esfuerzo computacional, y el orden de convergencia con respecto a la entrada N .

Teorema 4.6.2 Sea N la capacidad de buffer de todos los pares presentes en la red, y llamemos $T(N)$ al tiempo computacional necesario para evaluar la calidad $E(X_\pi)$. Si admitimos que tanto la cantidad de hormigas a utilizar como el número máximo de iteraciones ingresados son del orden de N , entonces el tiempo medio total para correr el Algoritmo Principal τ , es:

$$\tau = O(N^3T(N)) \quad (4.51)$$

Prueba. Estudiemos la cantidad de evaluaciones de calidad que figura en cada uno de los bloques *InicializarAristas*, *InicializarFeromonas*, *AplicarACO* y la fase de Búsqueda Local.

En *InicializarAristas* (Función 7) cada hormiga efectúa un ciclo, y luego se debe calcular la calidad del mismo para actualizar la distancia de todos los enlaces visitados en el ciclo. Entonces hay tantas evaluaciones de calidad como hormigas en la entrada. Sabemos que si las hormigas no repitieran enlaces, con $\frac{N}{2}$ hormigas bastaría para inicializar todos los enlaces del N -clic. Tomando $2N$ hormigas (o más genéricamente un orden lineal en N de hormigas a la entrada) determinamos que este bloque es lineal en la entrada N .

Centremos nuestra atención ahora en la Función *InicializarFeromonas*. Esta función recibe los miembros de la subfamilia, donde $I + J \leq N - 1$, y para cada miembro de la subfamilia se debe evaluar una vez la calidad. Definiendo $K = N - 1 - J - I$, la cantidad de soluciones en naturales de la anterior desigualdad coincide con la de soluciones naturales de la ecuación $I + J + K = N - 1$, que es igual a $C_2^{N+1} = \frac{N(N+1)}{2}$. Esto muestra que la aplicación de la función *InicializarFeromonas* requiere calcular en el orden de N^2 calidades.

Estudiemos la Función *AplicarACO*. En la misma cada hormiga realiza un ciclo aleatorio en función de las feromonas y distancias de cada enlace. Cada uno de estos ciclos requiere realizar una evaluación posterior de calidad, a los efectos de actualizar la feromona de los enlaces pisados. Si consideramos una cantidad de hormigas a la entrada del orden de N , este bloque tiene una cantidad de evaluaciones lineal con N .

Por último el bloque de Búsqueda Local es el que domina en cuanto a esfuerzo computacional refiere. En sólo una iteración debe calcular la calidad de todas las vecinas a una permutación dada. El cardinal de su vecindad es la cantidad de pares de elementos de una permutación de tamaño $N - 1$, es decir $C_2^{N-1} = \frac{(N-1)(N-2)}{2}$. En el peor de los casos, la búsqueda local termina habiendo finalizado el tope de iteraciones n . Si suponemos que elegimos este tope de iteraciones igual a N (más genéricamente del orden de N), resulta que la búsqueda local requiere un total de $N(N-1)(N-2)/2$ evaluaciones (más genéricamente de orden cúbico en N).

En conclusión, luego del análisis previo y bajo las hipótesis de cantidad de hormigas y número máximo de iteraciones lineal con N , la etapa de búsqueda local impone el mayor esfuerzo computacional, que es cúbico con la entrada N . Si el tiempo medio de una evaluación del sistema para un N dado es $T(N)$, entonces el tiempo total $\tau = O(N^3T(N))$.

QED

En conclusión, se debe realizar una cantidad del orden cúbico de evaluaciones de calidad para correr el Algoritmo Principal. Vale destacar que la búsqueda exhaustiva (o “fuerza bruta”) realiza un total de $(N - 1)!$ evaluaciones, no siendo polinomial con la entrada N . La Búsqueda Local propuesta es tratable computacionalmente, y el Algoritmo Principal ofrece un compromiso entre la calidad y el tiempo consumido.

4.7. Conclusiones

Se ha introducido en este capítulo un conjunto de estrategias de selección de piezas de gran riqueza y diversidad, donde su cardinal es factorial en la capacidad de los pares. La Propiedad de Aproximación de Estrategias es uno de sus importantes potenciales, que se evidencia en la concepción de un primer algoritmo, que intenta reflejar a vectores de probabilidad de ocupación de entrada, con continuidad y latencia similares a los mismos. El diseño del vector de entrada a este algoritmo es complejo, mas su aplicación sobre vectores particulares enseñan pautas para diseñar estrategias, y han permitido destacar un subconjunto de Subfamilias de estrategias de permutación.

A continuación se impone la necesidad de caracterizar la optimalidad de una estrategia de selección, que capture la calidad de experiencia del video streaming, vinculada a los parámetros fundamentales que son tiempo de buffering y continuidad en la reproducción. Se propone un COP y mecanismos de búsqueda de estrategias de permutación de gran calidad. Una distancia entre permutaciones y noción de estructura de vecindad permiten plantear un nuevo algoritmo basado en una búsqueda local. Hay dos bondades claves respecto al algoritmo primitivo definido antes del COP. La primera es que alcanza con diseñar una vez sola la entrada. La segunda es que la mejora en cada iteración de la permutación en la búsqueda local permite lograr óptimos locales independientemente de la medida de optimalidad considerada, siempre que el tope de iteraciones no limite.

Por último se propone un algoritmo más elaborado, al que se denominó “Algoritmo Principal”. Se halla una biyección natural entre un ciclo dirigido en un N -clique y una permutación, hecho que promueve a traducir el COP a un TSP asimétrico. Vistas las bondades de eficiencia computacional y adaptabilidad al modelo de TSP, se eligió una metaheurística basada en ACO, y complementada mediante las ventajas de mejora que ofrece la búsqueda local en una etapa final. Se aprovecha el conocimiento de la subfamilia de permutaciones obtenida mediante previa experiencia en el diseño de estrategias, como también variantes de las medidas de optimalidad y flexibilidad de elección de parámetros en ACO.

A efectos prácticos, se dispone luego de este capítulo de tres algoritmos que permiten obtener estrategias a llevar a una red real, de distinto grado de complejidad, calidad y eficiencia computacional. Si se decide aplicar sea el algoritmo primitivo o bien el basado en búsqueda local, se hace necesario definir de modo no tan goloso un vector de probabilidad a la entrada, el cual se desea reproducir mediante la estrategia que se obtiene a la salida. Por otra parte, si se prefiere aplicar el Algoritmo Principal el diseño se centra en los parámetros en ACO. Una vez

elegidos, el algoritmo está automatizado completamente, pues ahora no es necesario diseñar una probabilidad de ocupación.

Atendamos ahora los resultados obtenidos en cada algoritmo. El primero ya logra mejoras relativas respecto de las anteriores estrategias relevadas en la literatura (Rarest First, Greedy y Mixta), por lo que en aspecto teórico ya se aprecian contribuciones a partir del mismo. Por otro lado, el algoritmo basado en búsqueda local apendiza al algoritmo primitivo una fase de búsqueda local. Independiente de la medida de calidad aquí considerada, este segundo algoritmo va a obtener una calidad no menor al primero. Por último, algo muy destacado en este capítulo es las ventajas de la aplicación de ACO en el Algoritmo Principal, como se ha podido apreciar de los resultados finales. Estos muestran que el problema ha sido capturado apropiadamente mediante el modelo ATSP, y también que la metaheurística basada en ACO permitió encontrar resultados tanto mejores que los obtenidos mediante estrategias clásicas.

En el Capítulo 5 se va a realizar un análisis comparativo de varias de las estrategias mencionadas en este capítulo, ya sobre una red real. La tesis se enfoca a continuación en adaptar el modelo matemático a los efectos de capturar el dinamismo y heterogeneidad siempre presentes en una red de streaming P2P. Esto sería directamente aplicado en redes de streaming de video. Se utiliza la red GoalBit, concebida principalmente para la distribución de video-streaming en vivo a los usuarios, con alta calidad de experiencia.

Parte IV

DESEMPEÑO DE NUEVAS ESTRATEGIAS EN GOALBIT

Capítulo 5

Análisis de Performance en GoalBit

5.1. Introducción

Ya hemos estudiado en capítulos anteriores un modelo simple de cooperación entre pares. El mismo asume una topología estática donde todos los pares presentes en la misma son idénticos. Hemos profundizado en el diseño de estrategias de selección de piezas, asumiendo que todos los pares aplican esta misma estrategia de selección, y bajo un estado estacionario. En el Capítulo 4 hemos propuesto pautas para el diseño de estrategias de selección de piezas. Se ha destacado una subfamilia de estrategias de selección de piezas que presentan un pico, cuya posición permite dar mayores valores de continuidad, cuidando el tiempo de buffering. En este capítulo aplicaremos una estrategia de selección de piezas destacada del Capítulo 4 sobre la plataforma real GoalBit, contrastando su desempeño con las clásicas estrategias Greedy y Rarest First. A tales efectos se accede a una emulación de la red. Los tiempos de arribo y egreso de los pares a la red se han registrado de una difusión real de contenidos de video con GoalBit, y son datos de entrada de la emulación. El emulador de la red reproduce en tiempo real el proceso de comunicación entre pares y entre par y tracker de GoalBit, considerando todos y cada uno de los mensajes y estrategias de selección de pares que se intercambian realmente en esta red (detallados en el Capítulo 2).

La Sección 5.2 muestra un contraste entre las hipótesis asumidas en el modelo matemático de cooperación y los nuevos desafíos inherentes a esta plataforma real. Se destacará el dinamismo, la heterogeneidad y necesidad de tiempo real, presentes en redes P2P de streaming de video en vivo. A su vez veremos que es posible caracterizar la continuidad en la reproducción y latencia monitoreando la cantidad y tiempo medio de interrupciones, como también las duraciones medias de estas últimas.

La Sección 5.3 analiza el potencial de la emulación, a efectos de la comprensión de la verosimilitud de la información estadística y las medidas de desempeño que se pueden obtener de la misma. También se mencionan anteriores transmisiones reales sobre la red GoalBit, y qué información se ha recabado de las mismas. Se estudia a continuación el emulador desde el punto de vista de entrada - salida, para su utilización a nivel de usuario.

La Sección 5.4 describe el escenario de prueba. Se explica las propiedades del ordenador en el que se ha corrido la emulación, y los tiempos de procesamiento contabilizados en las pruebas. Se revisan los parámetros de diseño del cliente GoalBit, que son a su vez las entradas de la emulación. Finalmente se mencionan las instancias a ser aplicadas, atendiendo fundamentalmente los elementos que afectan directamente en la estrategia de selección de piezas.

La Sección 5.5 muestra los resultados obtenidos mediante la aplicación de las instancias comentadas en la Sección 5.4. Se realiza una comparación del desempeño entre las estrategias Greedy, Rarest First y nuevas estrategias de permutación, las últimas fruto de esta tesis.

Por último, la Sección 5.6 comprende las conclusiones de este capítulo.

5.2. Hipótesis asumidas VS Red real

Las redes de pares son de extrema complejidad, y en sus modelos matemáticos frecuentemente existen razones para criticar a cualquiera de sus hipótesis. Esta no es la excepción; mostraremos aquí la debilidad de las mismas en contraste con las redes reales. Por último se mencionan bondades de este modelo, que igualmente son razones para su consideración. La primera crítica al modelo matemático considerado es la concepción de la red como una topología cerrada. En la realidad ingresan y egresan constantemente pares a la red, lo que impone complejidad en el manejo de recursos, pues son dinámicos en el tiempo. La segunda crítica quiebra con la hipótesis de pares idénticos presentes en la red. Es claro que en redes P2P, que se montan sobre la infraestructura de Internet, los pares poseen anchos de banda muy variables, como también distinta capacidad de procesamiento. Esto tiene como consecuencia la distinta calidad de experiencia percibida por cada usuario, y variado grado de adaptabilidad de los pares a la red.

Un tercer elemento diferenciado entre el modelo y las redes reales de streaming de video es la hipótesis de sincronismo. La cooperación analizada en el modelo matemático se sostiene en el hecho que todos los pares reproducen la misma pieza en cada instante (si es que la disponen). A su vez, la consulta refiere a índices del buffer, entendidos estos globalmente, gracias al sincronismo. Lamentablemente, en las redes reales los pares reproducen la secuencia de video todos en distintos puntos. Esto trae serias consecuencias, que se enumeran a continuación.

La primera es que la comunicación entre pares puede fracasar desde el inicio, por el simple hecho de tener “ventanas desfasadas”. Los pares más avanzados en el tiempo esperan nuevas piezas que otros pares contactados aún desconocen su existencia. Esto significa que si un par reproduce piezas futuras en relación a la reproducción de otro, el primero nunca se va a poder beneficiar del último, fracasando así la cooperación por un problema de no sincronismo. Otro elemento muy aparejado con el asincronismo es el efecto parasitario. Un conjunto de pares frecuentemente de bajas capacidades de procesamiento y anchos de banda resultan rezagados respecto de la línea de reproducción de un par líder o actualizado (que reproduce piezas futuras en relación a cualquier otro par). Desafortunadamente, éstos no pueden cooperar a la gran proporción de pares que se hallan en tiempos de reproducción más avanzados. Lo úni-

co que pueden hacer es consumir recursos de la red para mantenerse conectados, rompiendo así con el esquema de cooperación característico basado en el mutuo beneficio de las redes P2P.

El principal potencial del modelo matemático estudiado es la captura de la cooperación entre los pares presentes en la red. A su vez se regocija en su simplicidad, elemento fundamental para tratar las complejas redes P2P. Otros elementos básicos siempre presentes en estas redes y también considerados son la fragmentación de la secuencia de video en piezas, la necesidad de coleccionar las piezas por parte de los pares y la continuidad del flujo de video, cualidades esenciales en el streaming de video. También, aunque simplificada, se encuentra la comunicación aleatoria entre los distintos pares de la red. Por último, el mecanismo de consulta entre pares es nuevamente simplificado pero consistente, pues en todos los casos se va a consultar por piezas faltantes, hecho destacado en este modelo.

Veremos más adelante un contraste entre distintas estrategias de selección de pares sobre la plataforma GoalBit, incluyendo entre estas a unas que han logrado altos desempeños bajo la perspectiva de este modelo matemático. De esta manera es posible comprender el potencial de capturar la cooperación y otros elementos básicos de redes de streaming P2P, o si la simplicidad del modelo es extrema.

5.2.1. Parámetros de desempeño en GoalBit

Los efectos de no sincronismo entre pares presentes en una red de streaming de video llevan a interesarse en nuevas medidas de desempeño. Particularmente, en la red GoalBit se observa que una vez que el par supera el tiempo de buffering inicial la descarga de piezas es en general continua, a menos que no se disponga de la siguiente pieza a ser reproducida. En tal caso, se aplica el concepto de re-buffering, que consiste en suspender la reproducción del video para obtener las piezas en el nuevo rango urgente. Entonces la medida de continuidad se puede concebir de dos maneras: la tradicional, referente a contabilizar el porcentaje de piezas reproducidas sobre el total (frecuentemente denominado Loss Rate del inglés) o la cantidad de re-buffering necesarios durante un período dado de transmisión de video. Puesto que las tasas de pérdidas de piezas es muy pequeña (por el hecho de suspender y descargar cuando falta una pieza) una medida de continuidad del video desde el punto de vista del usuario es el número de cortes, o cantidad de re-buffereos.

Por otra parte, la medida de latencia inicial se conserva como el tiempo de buffering, que es el tiempo que transcurre desde que un par se conecta a la red hasta que inicia la reproducción del video. Téngase presente que en la emulación existen pares que no inician la reproducción a pesar de ingresar a la red. Esto es un bug en el emulador que escapa a los objetivos de esta tesis (pues no refleja el comportamiento de la red real). Estos no se deben considerar para el estudio de latencia inicial. En conclusión, en GoalBit traducimos las medidas de desempeño a tres, a saber: latencia inicial, cantidad de interrupciones por re-buffering y tiempo medio de interrupciones.

5.3. Emulación de GoalBit

GoalBit fue desarrollado con la filosofía open source, en base al software libre VLC [80], cuya gran ventaja es la capacidad de reproducir contenidos multimedia en múltiples formatos y sistemas operativos. Trabajos seminales del equipo de desarrollo de GoalBit (inicialmente bajo el nombre de GolP2P) permitió la incorporación de diversos módulos que permitieron implantar la infraestructura de GoalBit sobre Internet. Inicialmente se contaba con un módulo que implementa un cliente BitTorrent mejorado (basado en el Enhanced CTorrent [30]) adaptado a streaming en vivo, otro módulo que realiza las funciones de Tracker y otros módulos destinados a la ejecución e interfaz gráfica amigable al usuario. Este desarrollo inicial posibilitó el despliegue inicial de GoalBit. Junto con el aumento del número de usuarios se tornó necesaria la medición del desempeño de esta red, fundamental para lograr mejoras sustanciales y escalabilidad. El grupo de desarrolladores ha creado nuevos módulos de mediciones a efectos de la comprensión del rendimiento global de GoalBit. Estos trabajos concluyen con un módulo gestor de simulaciones, varias adaptaciones del cliente BitTorrent y un módulo estadístico que permite capturar diversos parámetros de desempeño, como también una visualización gráfica que muestra el estado de reproducción de cada par perteneciente a la red en cada instante. Más recientemente, se ha desarrollado nuevos módulos orientados a lograr mayor verosimilitud con el proceso de distribución de contenidos de video en GoalBit en tiempo real. Estos ya son emulaciones de la red GoalBit.

En este apartado señalaremos las ventajas que presenta el uso del entorno de emulación de GoalBit frente a las simulaciones. Estudiaremos el potencial del entorno de emulación desde el punto de vista de un usuario: cualidades generales de la emulación, parámetros de control o de entrada y parámetros medidos o de salida. Finalmente se enumeran los pasos que un usuario debe realizar para completar una emulación exitosamente, para mostrar la simplicidad de su uso. El lector interesado en el funcionamiento de los módulos puede consultar la referencia [11].

5.3.1. Emulación VS Simulación de GoalBit

Si bien ambos tienen la finalidad de capturar el proceso real que ocurre en la distribución de video en la red GoalBit, se destacan ventajas del emulador sobre el simulador GoalBit, razones que motivan a utilizar aquí el primero.

La comunicación entre distintos nodos de la red es exitosamente lograda en ambos entornos. Ambos logran capturar la interacción entre pares y la comunicación par-tracker, pasando por cada etapa de definición de los mensajes. También en ambos casos se lee un archivo de instancia real sobre GoalBit y se procesa esta información en iteraciones.

Sin embargo, hay varias razones que destacan el uso del emulador. La primera es que en las simulaciones los pares no son ejecutados todos simultáneamente, hecho que sí se consigue en las emulaciones. Este elemento tiene importantes repercusiones. Una es que el tiempo de simulación va a ser mucho mayor que con emulaciones, dado que el tratamiento del intercambio de información en grupos de pares (y no todos a la vez) en las simulaciones lleva a transcurrir más tiempo de ejecución por iteración. Otra repercusión importante es la simplificación de implementación en las emulaciones: en estas es posible utilizar el reloj del sistema operativo.

Por su parte, las simulaciones dejan mensajes pendientes para siguientes iteraciones, y se debe aplicar un concepto de almacenamiento de mensajes (módulo denominado Outbox) a tales efectos. Además, en las simulaciones no se dispone de una red IP (la comunicación se basa en sockets), a diferencia del entorno de emulación. Esto genera dificultades adicionales en los algoritmos de control del ancho de banda de cada par.

Por último vale destacar que en ninguno de los entornos se agrega los efectos de red, pues al operar sobre un solo ordenador los retardos son despreciables. No obstante, el sistema operativo linux dispone de la herramienta llamada tc (Control de Tráfico [57]), que permite definir los efectos de red. En particular permite definir el retardo medio, su distribución de probabilidad y probabilidad de presencia de errores en ráfagas, entre otros.

Nuestros estudios comparativos con el agregado de la herramienta tc de linux, muestran que el efecto de red no es dominante ante la dinámica inherente a la red GoalBit. Por estas razones se opta en esta tesis por realizar el estudio comparativo entre distintas estrategias de selección de piezas con el entorno de emulación y sin pérdidas.

5.3.2. Emulación de GoalBit

A efectos de realizar una emulación de la plataforma GoalBit se requiere derechos de administrador sobre un sistema operativo Linux, y paquetes de software (módulos y programas de emulación), junto con información recabada de una experiencia real sobre la red GoalBit. Para el procesamiento estadístico es indispensable el módulo GoalBit Stats y la herramienta de desarrollo java.

El proceso de emulación imita el de montado de la red real GoalBit. Básicamente consiste en los siguientes 4 bloques, que se detallan a continuación:

- 1) Configurar los parámetros de entrada de la emulación.
- 2) Levantar al tracker.
- 3) Levantar una cierta cantidad de pares y correr la emulación.
- 4) Procesar el archivo de salida con el GoalBit Stats para obtener estadísticas de la emulación.

5.3.3. Bloque 1: Configurar parámetros de entrada

Los parámetros de entrada se pueden elegir alterando las líneas de dos archivos de extensión .conf, que se cargan en los programas. Uno corresponde al archivo broadcaster.conf, y permite elegir el número de broadcasters presentes en la red. El otro archivo de texto es goalbit-emu.conf, en el que se puede variar seis líneas:

- 1) *ruta*: se debe rellenar una línea que contenga la ruta donde se detallan los tiempos de conexión y desconexión de cada par. Este archivo de texto leído es normalmente extraído de información recabada de experiencia real en GoalBit.
- 2) *slice length*: recordemos que las piezas se componen de slice. Podemos elegir el tamaño de cada fragmento de pieza, en bytes.

- 3) *buffer size*: es el mínimo número de piezas consecutivas necesarias para que el par inicie la reproducción. Determina entonces la latencia inicial.
- 4) *urgent range size*: es el rango de piezas urgente próximo a la línea de reproducción. La estrategia de selección de piezas consiste en rellenar en primera instancia este rango en forma golosa (atendiendo siempre la pieza más próxima a la línea de reproducción). Una vez completado este sector de piezas se aplica una estrategia de selección de piezas en el sector no urgente del buffer, según los dos parámetros de entrada que se explican a continuación.

Comentario: Los siguientes dos parámetros son centrales en el diseño de la estrategia de selección de piezas a aplicar. Son los parámetros de control I y J referentes a la subfamilia de estrategias de permutación que presentan un pico. Se sugiere al lector la consulta del Capítulo 4 por detalles.

- 5) I : Una vez completado el sector urgente, se solicitan I piezas más lejanas a la línea de reproducción, dentro del tamaño de buffer.
- 6) J : Cuando se completa tanto el sector de buffer urgente como el de I piezas más lejanas a la línea de reproducción dentro del tamaño de buffer, se procede solicitando las J piezas no urgentes más próximas a la línea de reproducción.

En definitiva, los parámetros de entrada permiten elegir tamaños de buffer y fragmentos, como también el miembro de la subfamilia destacada de estrategias de permutación que presentan un pico. Se recuerda que la expresión de estrategia de permutación en función de los parámetros de control I y J viene dado por la siguiente expresión (definición 4.2.6):

$$\pi(i) = N - i, i = 1, \dots, I, \quad (5.1)$$

$$\pi(I + j) = j, j = 1, \dots, J \quad (5.2)$$

$$\pi(I + J + k) = \left\lfloor \frac{N + J - I}{2} \right\rfloor + \left\lfloor \frac{k}{2} \right\rfloor (-1)^{k+1}, \quad (5.3)$$

$$k = 1, \dots, N - I - J - 1,$$

Siendo N la diferencia entre el tamaño de buffer y el sector urgente. Obsérvese que se debe cumplir que $I + J < N$, para que la anterior expresión defina una estrategia de selección de piezas.

En definitiva, los parámetros de entrada determinan la estrategia de selección de piezas a aplicar (dentro de los miembros de la subfamilia), como también la escala de buffer y cantidad de distribuidores del video (broadcasters).

5.3.4. Bloque 2: Levantar el tracker

El primer nodo que se debe activar en la red es el tracker, pues es quien va a coordinar la comunicación entre los distintos nodos de la jerarquía de GoalBit (otorgando un ABI inicial a cada par normal, y monitoreando regularmente el estado de todos los pares de la red).

5.3.5. Bloque 3: Levantar los pares

Es posible seleccionar la cantidad de pares totales que arriban a la red. Esto se realiza coneccionando el registro real de la entrada para formar nuevos archivos de entrada con cantidades de pares deseados.

5.3.6. Bloque 4: Procesamiento estadístico

El programa principal retorna un archivo de extensión .log, cuyo formato se adapta a la entrada del módulo de GoalBit estadístico (GoalBitStats). Ya es posible abrir el archivo .log antes obtenido mediante la interfaz de GoalBitStats y visualizar y tratar estadísticamente la información. Las Figuras 5.1 y 5.2 muestran la información visual que se despliega al ejecutar GoalBitStats. La curva roja de la Figura 5.1 muestra el orden en el que arriban las piezas, donde cero es aquí la línea de reproducción (la línea recta por el origen es la estrategia Greedy). También se despliega el estado actual de cada par (ABI, pieza que se reproduce actualmente, cantidad de cortes y otros). La zona derecha se muestra con más detalle en la Figura 5.2. Para tres pares con sus correspondientes IPs, se ilustra la situación de reproducción actual. El par que actualmente está reproduciendo el video se muestra con el color azul (y la transición con el verde el instante de reproducción). En rojo se muestra el par que actualmente está descargando piezas, y con gris los pares desconectados.

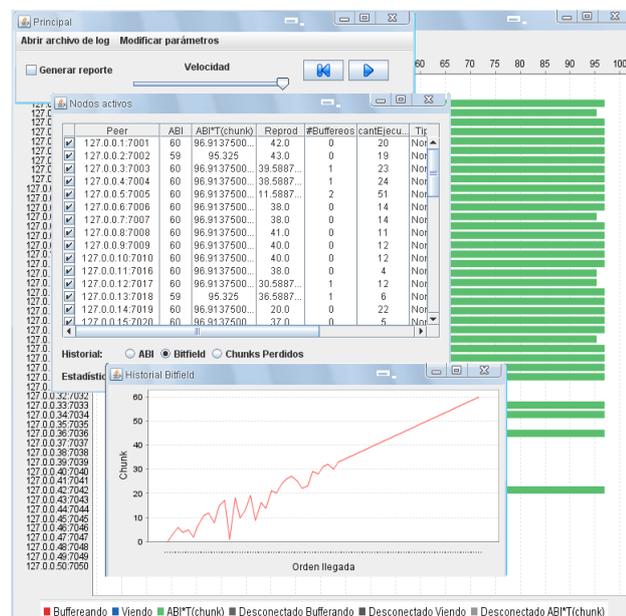


Figura 5.1: Pantalla de despliegue del GoalBit Stats. Se muestra información detallada del estado instantáneo de cada par.



Figura 5.2: Barras de reproducción de los pares.

5.4. Escenarios de emulación

En esta sección describiremos los casos de prueba que estudiaremos para comparar distintas estrategias de selección de piezas. Las emulaciones son posibles de realizar en un ordenador personal, o también en varios ordenadores simultáneamente. También se enumeran las características del ordenador utilizado para correr las emulaciones que se describen a continuación.

5.4.1. Propiedades del Ordenador

Las pruebas se realizaron en una máquina virtual cuyo sistema operativo huésped es Debian Lenny, Kernel 2.6.26. Fueron asignados 2 GB de RAM, y uno de los dos procesadores de la máquina residente, core 2 duo T8100 es de 2,10 GHz de capacidad.

5.4.2. Casos de prueba

Los archivos de entrada a goalbit-emu.conf contienen las horas de conexiones y desconexiones de los pares a la red. Esta información fue registrada en base a emisiones de partidos de fútbol uruguayos entre los años 2006 y 2007 sobre el servicio nacional de distribución de video AdinetTV [79]. Consideraremos a continuación 3 distintos archivos de entrada, en los que se varía el número de pares y tiempo de emulación. Para cada uno de estos archivos, se contrastan las estrategias de selección de piezas Greedy y Rarest First, contra la estrategia de la subfamilia con $I = 16$ y $J = 1$. A efectos comparativos, se mantienen constantes el resto de parámetros de entrada, variando únicamente I y J de modo de definir Rarest First ($I = N - 1$, $J = 0$), Greedy ($J = N - 1$, $I = 0$) o el miembro de la subfamilia de permutaciones con $I = 16$ y $J = 1$. El resto de los parámetros se eligen constantes, con los siguientes valores:

slice length: 65536 bytes.

buffer size: $N = 40$

urgent range size: 8

Para cada una de las tres estrategias de selección de piezas se aplican las 3 instancias indicadas en el Cuadro 5.1, donde cada iteración tiene una duración de diez segundos.

5.5. Resultados

Se han aplicado las tres anteriores estrategias de selección de piezas para cada una de las 3 instancias anteriores. A efectos de estudiar la latencia mostraremos para cada una de las

Instancia	Pares	Iteraciones
1	45	60
2	135	300
3	350	60

Cuadro 5.1: Tres instancias sobre las que se aplican las 3 estrategias de selección de piezas.

instancias y para cada par que ingresa a la red, la magnitud del tiempo de buffering inicial. Para contabilizar este tiempo de buffering se toma en cuenta únicamente aquellos pares que iniciaron la reproducción, valiéndonos del hecho que en la red real GoalBit todos los pares inician la misma.

La cantidad de re-buffereos realizados por cada par se grafican para cada instancia y cada par de la red, con la finalidad de comprender la continuidad y cantidad de interrupciones en cada estrategia de selección de piezas. Esto se complementa con un tercer gráfico que muestra el tiempo medio de buffering para cada par que presencia una interrupción (distinta de la inicial) y en cada instancia.

5.5.1. Instancia 1: 45 pares y 10 minutos

La Figura 5.3 muestra el tiempo de buffering inicial para cada uno de los pares que ingrean a la red. Los pares se enumeran según el orden de arribo a la red. La curva roja representa el número de iteración de llegada de cada par (cada iteración es de diez segundos). Por último las Figuras 5.4 y 5.5 muestran respectivamente la cantidad de buffereos totales y tiempo promedio de re-buffereos.

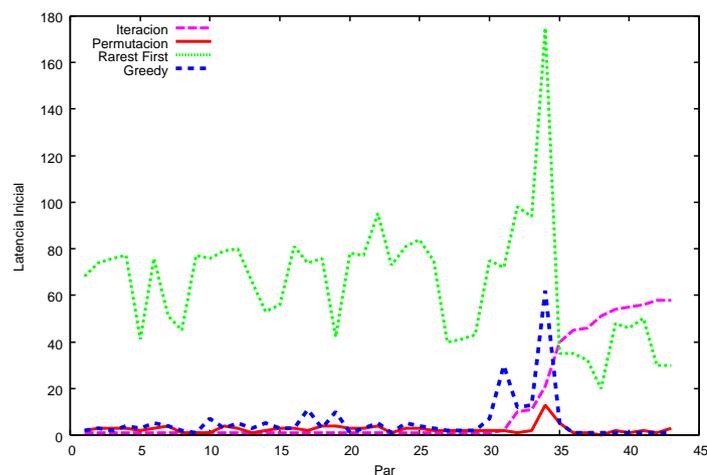


Figura 5.3: Tiempo de buffering inicial para distintas estrategias de selección de piezas

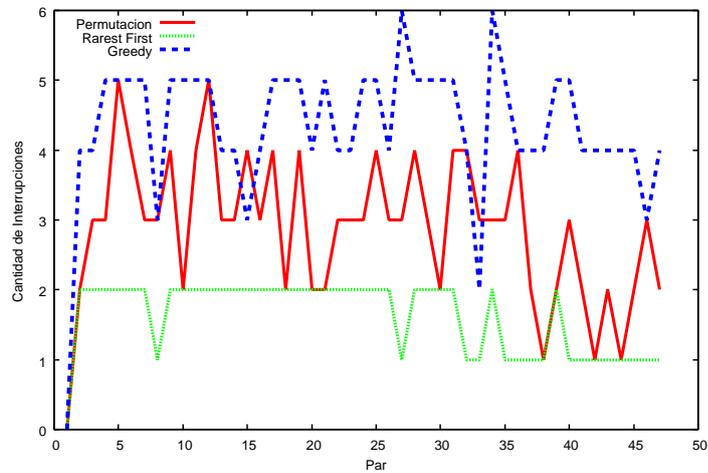


Figura 5.4: Cantidad de Interrupciones para distintas estrategias de selección de piezas

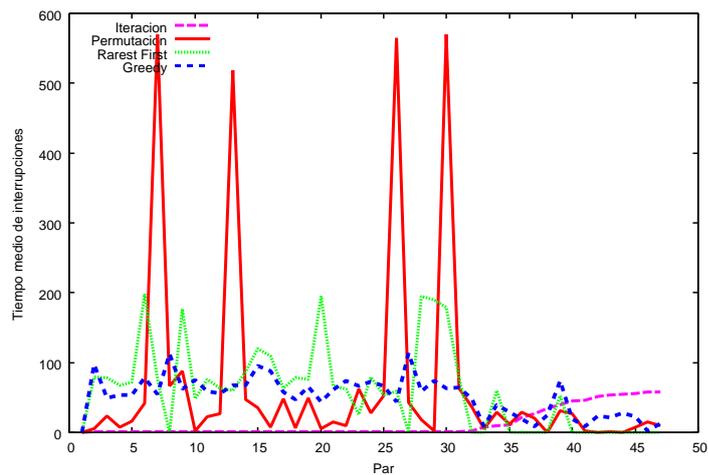


Figura 5.5: Tiempo medio de Interrupción para distintas estrategias de selección de piezas

5.5.1.1. Observaciones

Como se anticipaba, las latencias iniciales en Rarest First son muy altas, del orden de minutos. Esto comprueba nuestra deducción del Capítulo 3 que Rarest First posee latencias inaceptables a efectos de la distribución de video en tiempo real. Por su parte, tanto Greedy como la nueva estrategia de selección de piezas propuesta poseen tiempos de latencia iniciales de pocos segundos y comparables. Para esta cantidad de pares, la nueva propuesta ha superado a las clásicas en cuanto a latencia inicial refiere, tal como se muestra en la Figura 5.3. La cantidad de buffereos es superior en la nueva propuesta en relación a Rarest First. Sin embargo, esta estrategia de permutación posee salvo en el caso de 4 pares, tiempos promedio de re-buffereos menores que en las dos estrategias clásicas.

5.5.2. Instancia 2: 135 pares y 50 minutos

Las mismas corridas se aplican ahora para otro archivo de entrada, con 60 pares y 50 minutos de emulación. Los resultados se encuentran en las Figuras 5.6, 5.7 y 5.8.

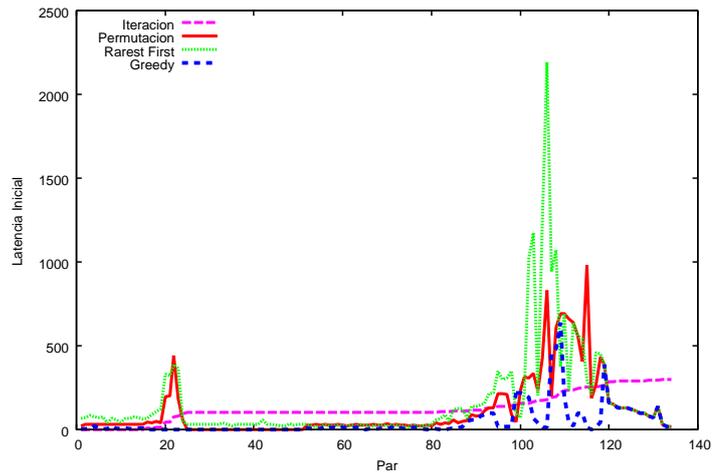


Figura 5.6: Tiempo de buffering inicial para distintas estrategias de selección de piezas

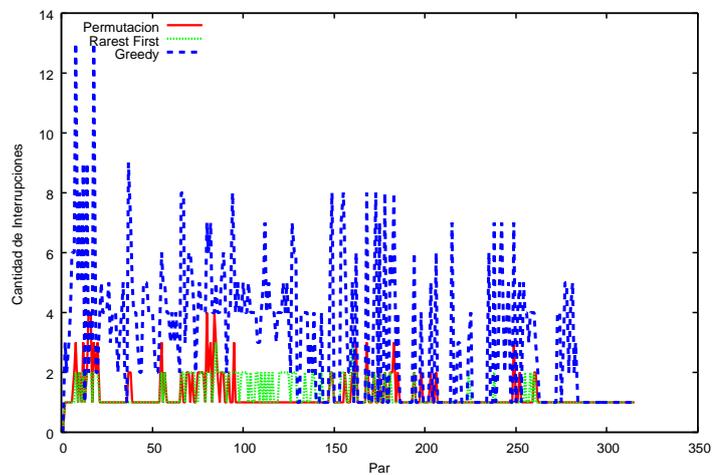


Figura 5.7: Cantidad de Interrupciones para distintas estrategias de selección de piezas

5.5.2.1. Observaciones

En este caso se presenta un gran número de pares que no han logrado iniciar la reproducción, en las tres estrategias. Esta emulación muestra inestabilidad y es dudosa la verosimilitud con la red en la segunda mitad (obsérvese las esperas iniciales de decenas de minutos, que un usuario humano sencillamente no esperaría). Con respecto al conjunto de pares que iniciaron

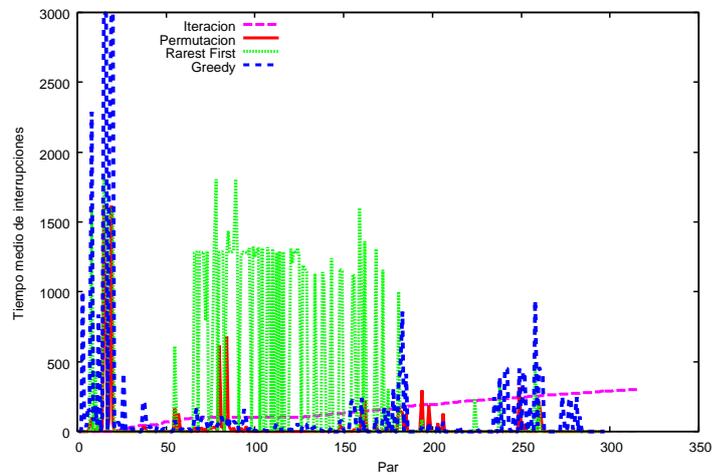


Figura 5.8: Tiempo medio de Interrupción para distintas estrategias de selección de piezas

la reproducción, el mejor desempeño en cuanto a continuidad lo presenta la nueva estrategia de permutación, con menos cantidad de interrupciones que las estrategias clásicas y tiempos promedio de re-buffereos muy reducidos, salvo para tres pares. Vale la pena destacar la alta cantidad de interrupciones que muestra la estrategia Greedy, elemento que muestra su no escalabilidad.

5.5.3. Instancia 3: 350 pares y 10 minutos

Los resultados de esta instancia se resumen en las Figuras 5.9, 5.10 y 5.11:

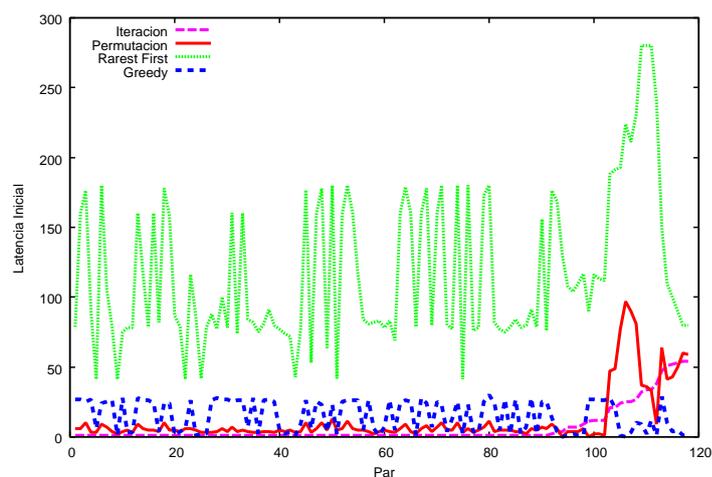


Figura 5.9: Tiempo de buffering inicial para distintas estrategias de selección de piezas

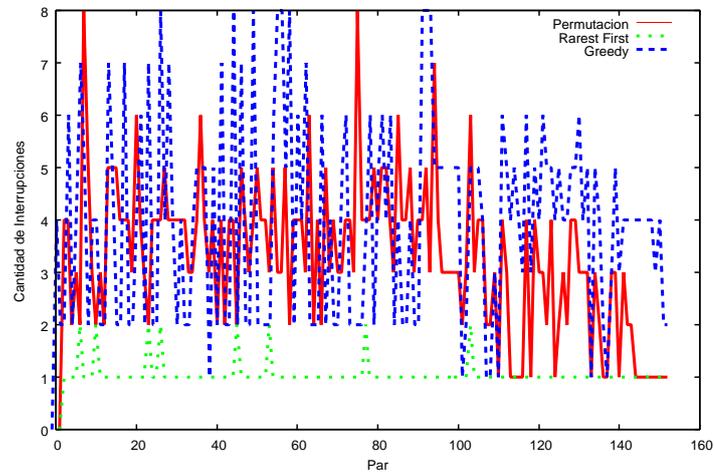


Figura 5.10: Cantidad de Interrupciones para distintas estrategias de selección de piezas

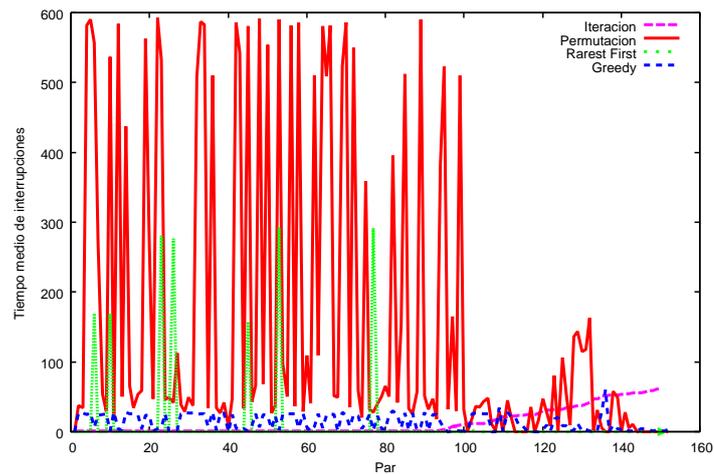


Figura 5.11: Tiempo medio de Interrupción para distintas estrategias de selección de piezas

5.5.3.1. Observaciones

La nueva propuesta presenta los tiempos de latencia inicial más reducidos, exceptuando los pares que ingresaron en el último tramo de la emulación. No obstante, la cantidad de interrupciones en este caso resultó comparable con las de Greedy, y el tiempo medio de cada interrupción es superior en la estrategia de permutación.

5.5.4. Resumen de Resultados

El Cuadro 5.2 muestra para cada una de las tres estrategias de selección de piezas e instancia, el promedio sobre todos los pares de la latencia inicial, cantidad de cortes y tiempos de los mismos. Greedy presenta en las tres instancias la mayor cantidad media de cortes, lo

que evidencia su no escalabilidad. Latencias iniciales medias de Rarest First es en todos los casos netamente superior que en el resto de las estrategias consideradas. Esto coincide con el análisis matemático realizado en el Capítulo 3 de Rarest First; que anticipa altas latencias de su aplicación.

Instancia	ESTRATEGIA	LATENCIA	CORTES	TIEMPO DE CORTES
1	Permutación	2.59	2.87	68.96
1	Rarest First	65	1.62	60.06
1	Greedy	5.67	4.34	51.81
2	Permutación	112.86	1.23	28.46
2	Rarest First	164.13	1.26	281.55
2	Greedy	44.62	3.18	109.23
3	Permutación	11.82	3.30	157.81
3	Rarest First	117.62	1.05	10.79
3	Greedy	15.09	3.93	13.28

Cuadro 5.2: Para cada instancia y estrategia de selección de piezas, se muestra en respectivas columnas la latencia inicial promedio, la cantidad promedio de cortes y la duración media de los mismos

5.6. Conclusiones

En este capítulo se ha podido hacer una comparación entre estrategias clásicas y un miembro de la subfamilia de permutaciones, conjunto destacado de estrategias de selección de piezas definido en el Capítulo 4. Actualmente GoalBit ofrece contenidos de video en tiempo real a gran escala en Internet, y también dispone de un simulador y emulador. Estos últimos desarrollos son fundamentales para estudiar la calidad percibida por un usuario, sin necesidad de desplegar un “contenido de prueba”, algo que no es deseable desde el punto de vista de los usuarios.

Las diferencias entre el modelo matemático introducido en el Capítulo 3 y la red real GoalBit son importantes, e incluyen la topología dinámica, el no sincronismo de los pares, desfasaje de sus ventanas deslizantes y distinción en la cantidad de recursos disponibles en cada par. Sus consecuencias son la presencia de pares parásitos, generación de interrupciones durante la reproducción del video, dificultad de intercambio mutuo de piezas (debido al desfasaje de ventanas), dificultad del inicio, entre otras.

Estas dificultades merecen concebir la continuidad contabilizando equivalentemente la cantidad de interrupciones y el largo medio de las mismas. Por su parte, la latencia inicial permanece midiéndose mediante el tiempo de buffering inicial, descartando pares que no logran adaptarse a la red (aquellos que nunca despliegan el contenido de video).

Importantes conclusiones se desprenden de los resultados. La primera es que los tiempos de buffering inicial de Rarest First son inaceptables para los requisitos de entrega de video en vivo. Esto refuerza el resultado de altas latencias explicado en el Capítulo 3, y además evidencia la no aplicación de BitTorrent en el streaming de video. No hay duda que son necesarias modifi-

caciones sustanciales de este protocolo para adaptarlo al streaming de video en vivo.

La segunda conclusión en base a los resultados es que la estrategia golosa (Greedy) es pobre en cuanto a continuidad en la reproducción, tal como se anticipa del Capítulo 3. En las tres instancias estudiadas, esta estrategia posee grandes cantidades de interrupciones y tiempos medios de las mismas. Por último, la nueva estrategia propuesta muestra latencias iniciales reducidas, mientras que la continuidad en la reproducción es superior a la de Greedy. Los resultados se asemejan a aquellos obtenidos matemáticamente a partir del estudio realizado en el Capítulo 4. Si bien en la práctica un usuario de GoalBit puede disfrutar de alta continuidad de video, existen pares que se atrasan y resultan parásitos dentro de la red. Frecuentemente son aquellos que poseen bajos anchos de banda o capacidad de procesamiento. Este efecto no sólo degrada la calidad de experiencia del usuario, sino que estos pares atrasados consumen recursos sin cooperar, quebrando con la filosofía P2P. Este es un problema abierto y no tratado en esta tesis, en la que la cooperación es optimista.

Por otra parte, recuérdese que las instancias aquí consideradas mantienen constantes el resto de parámetros distintos de los de control de estrategia de selección de piezas (I y J). Esto oculta la correlación existente entre la estrategia de selección de piezas con la de pares. Se está investigando actualmente el desempeño de distintas estrategias de selección de pares en GoalBit, basadas en P4P (Participación Proactiva del Proveedor en redes P2P). Trabajos futuros apuntan a definir una estrategia de selección de piezas en GoalBit estocásticas basadas en permutaciones. Una correcta integración entre ambas estrategias de selección (de piezas y de pares) permiten gran calidad de experiencia en el usuario final, que es el principal cometido de las redes de entrega de video en vivo.

Capítulo 6

CONCLUSIONES GENERALES

Las redes P2P son más eficientes que las redes tradicionales modalidad cliente-servidor. Esto ha permitido una mayor oferta de contenidos y su vez la masividad de la demanda.

La arquitectura de estas redes contrasta con el modelo cliente-servidor. Los pares participan activamente en la distribución del video, ofreciendo sus recursos para que todos logren obtener el contenido. La cooperación entre los pares es entonces un elemento crucial, y característico en estas redes.

Hay dos estrategias de diseño que afectan directamente el desempeño de la cooperación: la de selección de pares con los cuales se va a cooperar, y la del orden de selección de piezas. Esta tesis se centra en el diseño de nuevas estrategias de selección de piezas. En el Capítulo 1 se ha revisado la importancia de medir la calidad de experiencia de los usuarios en estas redes, que es el cometido final de diseño. La continuidad de reproducción del video y la latencia inicial son parámetros fundamentales que afectan a la calidad de experiencia. El Capítulo 2 presenta un protocolo de cooperación muy estudiado denominado BitTorrent, y la plataforma GoalBit, que es la primera red de streaming P2P de video en vivo que abre protocolos y código fuente y presenta actualmente gran difusión sobre Internet. Se han estudiado particularidades y desafíos de las redes de entrega de video en vivo, y el mecanismo de comunicación de pares y coordinación del tracker de GoalBit.

El Capítulo 3 presenta un modelo matemático simple de cooperación. Se repasan dos estrategias clásicas de selección de piezas denominadas Greedy (golosa) y Rarest First (primera más rara), mostrando sus deficiencias. A modo ilustrativo se introduce una estrategia de piezas Golosa Ponderada, que si bien no logra altas continuidades, es una primera aproximación en la búsqueda de estrategias de selección globalmente eficientes.

En el Capítulo 4 se investiga el diseño de estrategias de selección de piezas, y contiene las principales contribuciones de esta tesis. Se proponen nuevas estrategias de selección, como también un problema de optimización combinatoria que captura los parámetros fundamentales para distribuir video. Se define un gran número de nuevas estrategias, denominadas aquí “estrategias de permutación”, puesto que para cada permutación de los índices del buffer es posible tener una estrategia de selección de piezas. Esta gran cantidad de estrategias hace compleja la decisión de cuál de estas aplicar. Se estudian entonces propiedades de estas estrategias, medidas de optimalidad y nuevos algoritmos que automatizan la obtención de estrategias con

alta calidad. Se incluye una innovadora propuesta de resolución del problema de optimización combinatoria basado en optimización por colonia de hormigas y una búsqueda local. El espacio de permutaciones se enriquece de métrica, y se garantizan así óptimos locales del problema. Por último, en el Capítulo 5 se lleva a cabo una de las estrategias de permutación obtenida en GoalBit. Se contrasta con las clásicas estrategias de selección de piezas (Greedy y Rarest First). A pesar de la simplicidad del modelo matemático estudiado en el Capítulo 3, se observa la concordancia de conclusiones referentes a altas latencias con Rarest First y baja continuidad en la reproducción en Greedy. La nueva estrategia de permutación logra latencias iniciales menores que Rarest First y a su vez menos interrupciones que Greedy. No obstante, la calidad de experiencia registrada en todos los casos no muestra ser excelente, por lo que hay margen de mejoras y la investigación de nuevas estrategias de selección de piezas sigue en crecimiento.

PROBLEMAS ABIERTOS

A lo largo de esta investigación han surgido problemas tanto teóricos como prácticos que hasta la fecha no se les ha encontrado solución. A continuación se enumeran algunos de ellos.

Problemas Teóricos abiertos

Resolución de la recursión de Greedy y Rarest First

En el modelo introducido en el Capítulo 3 se caracteriza el vector de probabilidades de ocupación p tanto para Rarest First como para Greedy en forma recursiva. Hasta la fecha se desconoce una resolución explícita de estas recurrencias.

Ambos problemas son enteramente matemáticos. Se formulan de la siguiente manera: dados dos enteros positivos M (número de pares) y N (tamaño de buffer), hallar una expresión explícita para el vector p^{RF} que satisface la siguiente recursión:

$$\begin{aligned} p_1^{RF} &= \frac{1}{M} \\ p_{i+1}^{RF} &= p_i^{RF} + (1 - p_i^{RF})^2 p_i^{RF}, \forall i \in \{1, \dots, N-1\} \end{aligned}$$

Análogamente, el problema para Greedy consiste en expresar explícitamente el vector p^G dados M y N en la siguiente expresión:

$$\begin{aligned} p_1^G &= \frac{1}{M} \\ p_{i+1}^G &= p_i^G + (1 - p_i^G) p_i^G (1 - p_N^G + p_{i+1}^G - p_1^G), \forall i \in \{1, \dots, N-1\} \end{aligned}$$

La resolución de estos dos problemas permitirían en particular hallar expresiones explícitas para la continuidad y latencia en las estrategias clásicas, brindando mayor facilidad en el análisis y explicación más fundada de las causas de no escalabilidad de las mismas.

Hallar la Permutación óptima

Un problema más general que engloba a los anteriores es la de hallar una expresión explícita para una estrategia de permutación dada una permutación arbitraria de los primeros $N-1$ índices de buffer. Formalmente, dada una permutación $\pi : \{1, \dots, N-1\} \rightarrow \{1, \dots, N-1\}$

se quiere encontrar una expresión explícita para p_i , donde p_i cumple la siguiente recursión:

$$\begin{aligned} s_{\pi_1} &= 1 - \frac{1}{M} \\ p_1 &= \frac{1}{M} \\ s_{\pi_{i+1}} &= s_{\pi_i}(1 - p_{\pi_i}(1 - p_{\pi_i})), \forall i \in \{1, \dots, N - 2\} \\ p_{i+1} &= p_i + (1 - p_i)p_i s_i, \forall i \in \{1, \dots, N - 1\} \end{aligned}$$

La resolución de este problema no sólo permitiría resolver las recurrencias de Rarest First y Greedy (que se corresponden con las instancias de permutaciones inversa e identidad respectivamente), sino que lograría hallar una expresión explícita para la continuidad y latencia para cada permutación. Si bien se duda que sea posible la obtención de tal expresión explícita, esto cerraría el estudio de optimalidad sobre el modelo matemático estudiado, bajo cualquier medida de optimalidad. El sistema no lineal en estudio se ha resuelto mediante la aplicación del método de Newton-Raphson para sistemas. El conjunto de ecuaciones componentes de este sistema es infinitamente derivable, y el dominio de p_i y s_i son cubos; por tanto compacto y convexo. En caso de generar Newton-Raphson una contracción, converge con orden al menos cuadrático. Sin embargo, hasta la fecha no se ha demostrado que este método numérico sea convergente para este sistema no lineal, a pesar de haber convergido numéricamente en todas sus resoluciones, tomando como punto inicial el vector nulo.

Otro problema teórico abierto (que surgiría directo ante la resolución de los dos primeros) es el de determinar para qué par de enteros positivos M y N , la extensión de una consulta en Rarest First es menor en media que la de Greedy. A partir del análisis matemático del Capítulo 4, esto es equivalente a encontrar el subconjunto de pares naturales siguiente:

$$\left\{ (M, N) \in \mathcal{N} \times \mathcal{N} : p_N^{RF} + \frac{1}{M} < \frac{L^G + L^{RF}}{N} \right\},$$

donde L^G y L^{RF} representan las latencias en Greedy y Rarest First, bajo las respectivas probabilidades de ocupación p^G y p^{RF} .

Complejidad del COP propuesto

En esta tesis se ha introducido un problema de optimización combinatoria cuya complejidad hasta hoy es desconocida. Si bien hemos mostrado una traducción al problema TSP asimétrico, esta traducción no es una biyección. Actualmente se está trabajando para determinar la complejidad, cuestionándose la calidad de NP-Complejidad del COP de esta tesis. En caso de ser cierto este resultado y bajo el supuesto que $P \neq NP$, la eficiencia computacional cúbica en la entrada N obtenida por algoritmos del Capítulo 4, sería sobresaliente por su compromiso de calidad y tiempo computacional.

Nuevo modelo con hipótesis menos restrictivas

El modelo matemático simplificado de cooperación no incluye la pérdida de sincronismo, diferenciación de características (ancho de banda por ejemplo) de los pares de la red, ingreso y

egreso de pares y consideración de un rango de buffer urgente como otro parámetro de decisión. Está abierta la definición de un modelo matemático que contemple estos elementos de diseño, como también una extensión del estudio de esta tesis a efectos de adaptar el diseño a cualquier escala (cantidad de pares y otros parámetros de entrada relevantes).

Problemas Prácticos Abiertos

Si bien existe una gran cantidad de problemas prácticos abiertos en redes de streaming P2P de video en vivo, resaltemos aquí los relacionados a este trabajo; más específicamente al diseño de estrategias de selección de piezas en GoalBit. Si bien el modelo matemático de cooperación y el estudio teórico brinda nuevas estrategias, estas se reflejan en la práctica como pautas de diseño. Esto es así debido a la gran complejidad de diseño práctico, que está irremediamente ligado a estrategias de selección de pares y cualidades de la distribución (cantidad de broadcasters del video y superpares, sus anchos de banda, fallas en la distribución a nivel de red y otras).

Adaptación de las estrategias de selección a GoalBit

El estudio teórico realizado en esta tesis brinda pautas de diseño, y es un indicador de estrategias de buena calidad. No obstante, su aplicación por sí sola no logra amplias calidades de experiencia, y es necesario un análisis más detallado del proceso global de cooperación para cada red en particular.

Para el caso de la red GoalBit, hay al menos dos elementos de diseño que afectan significativamente el desempeño global de la red, y se centralizan a nivel del tracker. Uno es el de estrategia de selección de pares. Más precisamente, el proceso por el cual se selecciona una lista de pares activos en la red. Esta lista se otorga a un par que ingresa a la misma, con los cuales iniciará la comunicación. La cantidad de miembros presentes en la misma juega un rol crucial: intuitivamente si son pocos, la cooperación va a fracasar ante la escasa diversidad de piezas en conjunto. En caso contrario, listas con excesivo número de pares tiende no sólo a cargar la red y aumentar los tiempos de establecimiento de la comunicación, sino también recaen en una mayor complejidad en el control de información por parte del tracker.

Otro elemento que afecta el desempeño de la red es la asignación del punto inicial de reproducción del video. Recuérdese que una vez que un par ingresa a la red, el tracker brinda un ABI (índice de buffer activo) a partir del cual el par puede iniciar la reproducción. Este punto juega un rol crucial en la determinación de miembros parásitos de la red, pues si se asigna ABIs muy bajos, el par no va a poder cooperar. Por otra parte, la asignación de ABIs muy altos van a hacer que el par que ingresa pueda ser beneficiado por muy pocos miembros de la red. Por ahora se retorna a cada nuevo par de la red el ABI del broadcaster, siendo esta política posiblemente mejorable.

En este panorama es que se aplica el contraste entre las estrategias clásicas y la nueva propuesta. Si se desea mantener modelos simples de cooperación, es interesante proceder mediante una aleatorización de las estrategias de permutación. Esfuerzos futuros se pueden dirigir a definir distribuciones de probabilidades (dependientes de la distribución empírica de posicionamiento del ABI) que apliquen distintas permutaciones del orden de consulta de piezas con distintas

probabilidades. Un elemento importante en esta dirección es el de definir una medida de calidad de experiencia en base a un conteo de la cantidad de interrupciones, tiempo medio de estas y latencia inicial, en paralelismo con los dos parámetros destacados en esta tesis.

Si bien aquí no se va a abordar un problema de tan amplio espectro como el anterior, un ordenamiento posible de trabajo orientado a mejorar el desempeño de GoalBit es el siguiente:

- 1 Determinar empíricamente la probabilidad de éxito en el establecimiento de una comunicación. Esto dejaría en evidencia la necesidad de incrementar o reducir el tamaño de la lista a asignar a un nuevo par por parte del tracker.
- 2 Determinar el número óptimo de tamaño de lista a brindar por el tracker a nuevos pares. Esto se puede hacer en base al estudio anterior, y considerando a la vez sugerencias de trabajos sobre redes con similares características [77].
- 3 Analizar la distribución de probabilidad del ABI de los pares con el tiempo, en relación con el promedio.
- 4 Proponer una medida de optimalidad que contemple interrupciones, tiempo de latencia inicial y tiempo medio de interrupción.
- 5 Definir nuevas estrategias de permutación aleatorizadas, en base al conocimiento adquirido en pasos anteriores y posiblemente a una adaptación de parámetros a optimizar según el modelo y máxima verosimilitud.

Bibliografía

- [1] 3GPP Specifications Home Page. <http://www.3gpp.org/specs/specs.htm>, 2007.
- [2] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Robust tcp congestion recovery. *21st International Conference on Distributed Computing Systems, 2001.*, 2001.
- [3] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Tcp-friendly congestion control for highspeed network. *SAINT 2007: International Symposium on Applications and the Internet.*, 2007.
- [4] H. Al-Zubaidy and T. Omari. Red performance evaluation using stochastic modelling and fluid-based analysis. In *Canadian Conference on Electrical and Computer Engineering, CCECE 2006*, pages 1413–1418, Ottawa, CANADA, 2006. IEEE Computer Society.
- [5] Alcatel-Lucent Home. <http://www.alcatel-lucent.com/>, 2007.
- [6] Shahzad Ali, Anket Mathur, and Hui Zhang. Measurement of commercial peer-to-peer live video streaming. In *In Proc. of ICST Workshop on Recent Advances in Peer-to-Peer Streaming*, Weaterloo, Canadda, 2006.
- [7] ANSI Standard T1.801.03. Digital transport of one-way video telephony signals, January 1996.
- [8] Daniel C. Asmar, Ahmad Elshamli, and Shawki Areibi. A comparative assessment of aco algorithms within a tsp environment. In *the 4th International Conference on Engineering Applications and Computational Algorithms*, 2005.
- [9] Robert Axelrod. *The Evolution of Cooperation*. 1984.
- [10] P. Baccichet, T. Shierl, T. Wiegand, and B. Girod. Low-delay peer-to-peer streaming using scalable video coding. november 2007.
- [11] A. Barrios, M. Barrios, J. Comas, and P. Perdomo. Análisis y evaluación del rendimiento de la plataforma P2P Goalbit, 2009.
- [12] A. Basso, I. Dalgic, F. Tobagi, and C. Lambrecht. Study of mpeg-2 coding performance based on a perceptual quality metric. In *Proceedings of PCS'96*, pages 263–268, March 1996.

- [13] BBC: iMP website. <http://www.bbc.co.uk/imp/>, 2007.
- [14] R. Beckers, J.L. Deneubourg, and S. Goss. Trails and U-turns in the selection of the shortest path by the ant *lasius niger*. *Journal of Theoretical Biology*, 159:397–415, 1992.
- [15] Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, and Bernd Pollermann. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1992.
- [16] H. Bersini, M. Dorigo, S. Langerman, G. Seront, and L. M. Gambardella. Results of the first international contest on evolutionary optimization. In *International Conference on Evolutionary Computation*, pages 611–615. IEEE Press, 1996.
- [17] María Elisa Bertinat, Daniel De Vera, Darío Padula, Franco Robledo, Pablo Rodríguez-Bocca, and Pablo Romero. Systematic procedure for improving continuity and latency on a p2p streaming protocol. In *To appear in Proceedings of IEEE Latin-American Conference on Communications (LatinCom 2009)*. IEEE, 2009.
- [18] María Elisa Bertinat, Daniel De Vera, Darío Padula, Franco Robledo, Pablo Rodríguez-Bocca, Pablo Romero, and Gerardo Rubino. A cop for cooperation in a p2p streaming protocol. In *To appear in Proceedings of International Conference in Ultra Modern Telecommunications (ICUMT 2009)*. IEEE, 2009.
- [19] María Elisa Bertinat, Daniel De Vera, Darío Padula, Franco Robledo, Pablo Rodríguez-Bocca, Pablo Romero, and Gerardo Rubino. Goalbit: The first free and open source peer-to-peer streaming network. In *To appear in Proceedings of the 5th international IFIP/ACM Latin American conference on Networking*, New York, USA, 2009. ACM.
- [20] Bittorrent home page. <http://www.bittorrent.org>, 2007.
- [21] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. Technical report, Université Libre de Bruxelles, IRIDIA, 2001.
- [22] Christian Blum. Ant colony optimization: Introduction and recent trends. *Physics of life Reviews*, 2:353–373, october 2005.
- [23] T. Bonald, M. May, and J. Bolot. Analytic evaluation of red performance. In *INFO-COM 2000: Joint Conference of the IEEE Computer and Communications Societies*, pages 1415–1424, Washington, DC, USA, 2000. IEEE Computer Society.
- [24] Jean-Ives Le Boudec. Some properties of variable length packet shapers. *IEEE/ACM Transactions on Networking*, 10(3):329–337, 2002.
- [25] E. Buyukkaya, M. Abdallah, and R. Cavagna. Vorogame: A hybrid p2p architecture for massively multiplayer games. In *Consumer Communications and Networking Conference, CCNC 2009*, pages 1–5, January 2009.
- [26] Cachelogic home page. <http://www.cachelogic.com>, 2007.

- [27] Wu Chehai, Lu Xianliang, and Duan Hancong. Analysis of content availability optimization in bittorrent. In *ICHIT 2006: International Conference on Hybrid Information Technology*, pages 525–529, Chinese Academy of Science, 2006. IEEE Computer Society.
- [28] Z. Chen, H. Wang, B. Yang, and Y. Chen. A novel method of p2p hosts detection based on flexible neural tree. In *ISDA '06: Conference on Intelligent Systems Design and Applications*, pages 556–561. IEEE, 2006.
- [29] Cisco Systems, Inc. <http://www.cisco.com/>, 2007.
- [30] Cliente CTorrent. <http://sourceforge.net/projects/ctorrent/>, 2004.
- [31] Bram Cohen. Incentives build robustness in bittorrent. *www.bramcohen.com*, 1:1–5, May 2003.
- [32] CoolStreaming home page. <http://www.coolstreaming.us>, 2007.
- [33] Charles D. Cranor, Matthew Green, Chuck Kalmanek, David Shur, Sandeep Sibal, Jacobus E. Van der Merwe, and Cormac J. Sreenan. Enhanced streaming services in a content distribution network. *IEEE Internet Computing*, 5(4):66–75, 2001.
- [34] C. Dale and J. Liu. A measurement study of piece population in bittorrent. *IEEE Global Telecommunications Conference, GLOBECOM 2007*, pages 405–410.
- [35] DCIA Activities. P4P Working Group (P4PWG). <http://www.dcia.info/activities/#P4P>, 2009.
- [36] Jean Paul Delahaye and Philippe Mathieu. Complex strategies in the iterated prisoner's dilemma. In *Chaos and Society*, pages 283–292. IOS Press, 1995.
- [37] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, DEI Politecnico de Milano, Italia, 1992.
- [38] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Artificial ants as a computational intelligence technique. Technical Report 23, Institut de Recherches Interdisciplinaires, Université Libre de Bruxelles, september 2006.
- [39] Marco Dorigo, Eric Bonabeau, and Guy Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, pages 851–871, 2000.
- [40] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [41] Marco Dorigo and Thomas Stutzle. *Ant Colony Optimization*. MIT Press, 2004.
- [42] Haibin Duan, Guanjun Ma, and Senqi Liu. Experimental study of the adjustable parameters in basic ant colony optimization algorithm. *IEEE Congress on Evolutionary Computation*, 1(1):149–156, 2007.

- [43] C. Bernhardt E. Biersack, F. Thiesse. Constant data length retrieval for video servers with variable bit rate streams. In *International Conference on Multimedia Computing and Systems*, pages 151–155, Sophia Antipolis, France, 1996. IEEE Proceedings.
- [44] N. Fonseca and M. Crovella. Bayesian packet loss detection for tcp. In *INFOCOM 2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1826–1837. IEEE Press, 2005.
- [45] E. Gelenbe and J.M. Fourneau. Random Neural Networks with Multiple Classes of Signals. *Neural Computation*, 11(3):953–963, 1999.
- [46] E. Gelenbe, V. Koubi, and F. Pekergin. Dynamical random neural network approach to the traveling salesman problem. *Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, 2:630–635, 17-20 Oct 1993.
- [47] GoalBit - The First Free and Open Source Peer-to-Peer Streaming Network. <http://goalbit.sf.net/>, 2008.
- [48] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. Insights into p2p live: A measurement study of a large-scale p2p iptv system. In *In Proc. of IPTV Workshop, International World Wide Web Conference*, 2006.
- [49] Shay Horovitz and Danny Dolev. LiteLoad: Content unaware routing for localizing P2P protocols. In *Proceeding of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS'08)*, pages 1–8, Miami, USA, April 2008.
- [50] P. Hurley, M. Kara, J-Y. Le Boudec, and P. Thiran. ABE: Providing a Low-Delay Service within Best Effort. *IEEE Network Magazine*, 15(3), May 2001.
- [51] B. Hölldobler and E. O. Wilson. *Journey to the Ants: A Story of Scientific Exploration*. Harvard University Press, Cambridge, 1994.
- [52] ITU-R Recommendation BT.500-11. Methodology for the subjective assessment of the quality of television pictures, June 2002.
- [53] jumpTV Home page. <http://www.jumptv.com/>, 2007.
- [54] Arnaud Legout, G. Urvoy-Keller, and P. Michiardi. Rarest first and choke algorithms are enough. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 203–216, New York, NY, USA, 2006. ACM.
- [55] Dongmei Lin, Xiangbin Wu, and Dong Wang. Exact heuristic algorithm for traveling salesman problem. In *The 9th International Conference for Young Computer Scientists*, pages 9–13, Washington, DC, USA, 2008. IEEE Computer Society.
- [56] Yao Liu, Lei Guo, Fei Li, and Songqing Chen. A case study of traffic locality in internet p2p live streaming systems. In *International Conference on Distributed Computing Systems, ICDCS 2009*, pages 423–432, June 2009.

- [57] Man de Linux. <http://www.kernel.org/doc/man-pages/>, 2005.
- [58] Nicholas Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, september 1949.
- [59] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulations and traveling salesman problems. *ACM*, 7:326–329, 1960.
- [60] Allen K. Miu, John Apostolopoulos, Wai tian Tan, and Mitchell Trott. Low-Latency Wireless Video Over 802.11 Networks Using Path Diversity. In *IEEE ICME 2003*, Baltimore, USA, July 2003.
- [61] S. Mohamed and G. Rubino. A Study of Real-time Packet Video Quality Using Random Neural Networks. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(12):1071–1083, December 2002.
- [62] S. Mohamed, G. Rubino, and M. Varela. Performance evaluation of real-time speech through a packet network: a Random Neural Networks-based approach. *Performance Evaluation*, 57(2):141–162, May 2004.
- [63] msnTV website. <http://www.msntv.com>, 2007.
- [64] MyTVPal Home page. Free Internet TV. <http://www.mytvpal.com/>, 2007.
- [65] L. Nielsen. A Neural Network Model for Prediction of Sound Quality. Technical report, Technical University of Denmark, 1993.
- [66] Temel Oncan, Y. Kuban Altynel, and Gilbert Laporte. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research.*, 36(3):637–654, 2009.
- [67] P4P - Enable ISP and P2P to Work Together. <http://www.openp4p.net/>, 2009.
- [68] Dongsheng Peng, Weidong Liu, and Chuang Lin. Enhancing tit-for-tat strategy to cope with free-riding in unreliable p2p networks. In *Third International Conference on Internet and Web Applications and Services*, pages 336–341, Tsinghua University, Beijing, 2008. IEEE Computer Society.
- [69] M. Pinson and S. Wolf. A new standarized method for objectively measuring video quality. *IEEE Transactions on Broadcasting.*, 50(3):312–322, september 2004.
- [70] PPLive Home page. <http://www.pplive.com>, 2007.
- [71] PPStream home page. <http://www.ppstream.com/>, 2007.
- [72] Martin Varela Rico. *Pseudo-subjective Quality Assessment of Multimedia Streams and its Applications in Control*. PhD thesis, INRIA/IRISA, Univ. Rennes I, Rennes, France, nov 2005.

- [73] Matei Ripeanu. Peer-to-peer architecture case study: Gnutella network. In *Proceedings of the First International Conference on Peer-to-Peer Computing, 2001*, pages 99–100, August 2001.
- [74] Pablo Rodríguez-Bocca. *Quality-centric design of Peer-to-Peer systems for live-video broadcasting*. PhD thesis, INRIA/IRISA, Université de Rennes I, Rennes, France, april 2008.
- [75] Pablo Rodríguez-Bocca, Hector Cancela, and Gerardo Rubino. Perceptual quality in P2P video streaming policies. In *50th IEEE Global Telecommunications Conference (GLOBECOM'07)*, Washington DC, United States, 26-30 November 2007.
- [76] J. Rosenberg and H. Schulzrinne. Timer reconsideration for enhanced rtp scalability. In *INFOCOM '98: Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1413–1418. Bell Labs, USA, 1998.
- [77] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari. Will iptv ride the peer-to-peer stream? *Communications Magazine, IEEE*, 45:86–92, 2007.
- [78] Siemens AG. <http://www.siemens.com/>, 2007.
- [79] Sitio de AdinetTV. <http://www.adinetv.com.uy/>, 2006.
- [80] Sitio Web de Video Lan Client. <http://www.videolan.org/>, 2009.
- [81] SopCast - Free P2P internet TV. <http://www.sopcast.org>, 2007.
- [82] Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang. An analysis of live streaming workloads on the internet. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 41–54, New York, NY, USA, 2004. ACM Press.
- [83] TVAnts home page. <http://cache.tvants.com/>, 2007.
- [84] TVUnetworks home page. <http://tvunetworks.com/>, 2007.
- [85] C. J. van den Branden Lambrecht. *Perceptual Models and Architectures for Video Coding Applications*. PhD thesis, EPFL, Laussane, Swiss, 1996.
- [86] Vijay V. Vazirani. *Approximation Algorithms*. Springer, New York, 2003.
- [87] Li-Ying Wang, Jie Zhang, and Hua Li. An improved genetic algorithm for tsp. In *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, pages 925–928, Washington, DC, USA, 2007. IEEE Computer Society.
- [88] S. Wee, W. Tan, J. Apostolopoulos, and S. Roy. System design and architecture of a mobile streaming media content delivery network (msdcnd). Technical report, Streaming Media Systems Group, HP-Labs, 2003.

- [89] Chi-Jen Wu, Cheng-Ying Li, and Jan-Ming Ho. Improving the download time of bittorrent-like systems. In *IEEE International Conference on Communications 2007 (ICC 2007)*, pages 1125–1129, Glasgow, Scotland, June 2007.
- [90] Haiyong Xie, Arvind Krishnamurthy, Avi Silberschatz, and Yang Richard Yang. P4P: Proactive Provider Participation for P2P. In *P4PWG Whitepaper*, march 2007.
- [91] Haiyong Xie, Yang Richard Yang, Arvind Krishnamurthy, Yanbin Liu, and Avi Silberschatz. P4P: Portal for (P2P) Applications. In *Proceeding of the conference of the Special Interest Group on Data Communications, a special interest group of the Association for Computing Machinery (SIGCOMM'08)*, Seattle, USA, August 2008.
- [92] Susu Xie, Gabriel Y. Keung, and Bo Li. A measurement of a large-scale peer-to-peer live video streaming system. In *ICPPW '07: Proceedings of the 2007 International Conference on Parallel Processing Workshops (ICPPW 2007)*, page 57, Washington, DC, USA, 2007. IEEE Computer Society.
- [93] youTube website. <http://www.youtube.com/>, 2007.
- [94] Guomin Zhang, Changyou Xing, and Ming Chen. A distributed multimedia cdn model with p2p architecture. In *International Symposium on Communications and Information Technologies, ISCIT 2006*, pages 152–156, October 2006.
- [95] X. Zhang, J. Liu, and B. Li. On large scale peer-to-peer live video distribution: Coolstreaming and its preliminary experimental results. In *IEEE International Workshop on Multimedia Signal Processing (MMSp'05)*, Washington, DC, USA, 2005. IEEE Computer Society.
- [96] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Tak-Shing Peter Yum. Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming. In *IEEE Conference on Computer Communications (INFOCOM'05)*, Washington, DC, USA, 2005. IEEE Computer Society.
- [97] Yipeng Zhou, Dah Ming Chiu, and J.C.S. Lui. A Simple Model for Analyzing P2P Streaming Protocols. In *Proceeding of the IEEE International Conference on Network Protocols (ICNP'07)*, pages 226–235, Beijing, China, October 2007.

Índice de figuras

1.1. Etapa de entrenamiento de PSQA.	21
1.2. Aplicación de PSQA.	21
2.1. Arquitectura GoalBit. Los 4 componentes de la red son el broadcaster, los superpares, los pares y el tracker.	30
2.2. Buffer Activo e índice del Buffer Activo (ABI).	30
2.3. Comunicación exitosa entre pares. La secuencia continúa con mensajes WINDOWUPDATE y PIECE intercalados. Nótese que cada vez que A obtiene una pieza, avisa mediante un HAVE a los restantes pares que puedan beneficiarse con la misma (B no se incluye en este grupo). A su vez A mantiene la comunicación con otros pares y el tracker.	39
2.4. Estrategia de selección de piezas.	41
3.1. Modelo de buffer en cada par. La posición 1 representa la pieza de video más nueva en la red y la posición N la siguiente pieza a ser desplegada. Obsérvese que todos los pares están sincronizados en el consumo de buffer.	46
3.2. Orden de consulta en las estrategias Greedy y Rarest First. Rarest First comienza la consulta en el índice 1 y avanza hacia la dirección de reproducción, incremental en índices. Por su parte Greedy comienza en el índice $N - 1$ y continúa la consulta alejándose de la línea de reproducción	50
3.3. Probabilidades de ocupación de las estrategias clásicas para 1000 pares y tamaño de buffer 40.	51
3.4. Contraste entre las estrategias Mixta, Greedy y Rarest First, para $M = 1000$, $N = 40$ y $m = 10$. La estrategia Mixta logra un interesante compromiso entre latencia y continuidad.	59
4.1. Probabilidad de ocupación segmentada	68
4.2. Estrategia ideal para la probabilidad de ocupación segmentada	69
4.3. Estrategia de permutación para el objetivo segmentado	69
4.4. Probabilidad de ocupación en la salida, asociada a la entrada segmentada	70
4.5. Evolución de la continuidad	70
4.6. Evolución de la latencia	71
4.7. Probabilidad de ocupación para cada estrategia	71
4.8. Comparación de la Continuidad entre diferentes estrategias	72
4.9. Comparación de la Latencia entre diferentes estrategias	72

4.10. Representación gráfica de niveles de calidad constante para ambos problemas de programación matemática (plano $p_1 = \frac{1}{M}$)	79
4.11. Salida del Algoritmo 2. Se observa la prioridad dada a la continuidad, penalizando la latencia.	90
4.12. Comparación entre distintas estrategias	99
5.1. Pantalla de despliegue del GoalBit Stats. Se muestra información detallada del estado instantáneo de cada par.	111
5.2. Barras de reproducción de los pares.	112
5.3. Tiempo de buffering inicial para distintas estrategias de selección de piezas	113
5.4. Cantidad de Interrupciones para distintas estrategias de selección de piezas	114
5.5. Tiempo medio de Interrupción para distintas estrategias de selección de piezas	114
5.6. Tiempo de buffering inicial para distintas estrategias de selección de piezas	115
5.7. Cantidad de Interrupciones para distintas estrategias de selección de piezas	115
5.8. Tiempo medio de Interrupción para distintas estrategias de selección de piezas	116
5.9. Tiempo de buffering inicial para distintas estrategias de selección de piezas	116
5.10. Cantidad de Interrupciones para distintas estrategias de selección de piezas	117
5.11. Tiempo medio de Interrupción para distintas estrategias de selección de piezas	117

