



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

ALGORITMOS EFICIENTES PARA JUEGOS DE STACKELBERG CON DEFENSORES
DESCENTRALIZADOS

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES

HUGO FERNANDO NAVARRETE ECHEVERRÍA

PROFESOR GUÍA:
FERNANDO ORDOÑEZ PIZARRO

MIEMBROS DE LA COMISIÓN:
DENIS SAURÉ VALENZUELA
RICHARD WEBER HAAS

Este trabajo ha sido parcialmente financiado por
CONICYT-PCHA/MagísterNacional/2015 - 22152053

Powered@NLHPC: Esta investigación fue parcialmente apoyada por
la infraestructura de supercómputo del NLHPC (ECM-02)

SANTIAGO DE CHILE
2018

RESUMEN DE LA TESIS
PARA OPTAR AL GRADO DE
MAGÍSTER EN GESTIÓN DE OPERACIONES
POR: HUGO FERNANDO NAVARRETE ECHEVERRÍA
FECHA: AGOSTO 2018
PROF. GUÍA: FERNANDO ORDOÑEZ PIZARRO

ALGORITMOS EFICIENTES PARA JUEGOS DE STACKELBERG CON DEFENSORES
DESCENTRALIZADOS

Uno de los desafíos importantes que enfrenta un grupo de defensores corresponde a su coordinación con el objetivo de poder brindar una mayor protección al sistema que defienden. En este trabajo, se estudia el desarrollo de algoritmos eficientes y garantías de optimalidad para un modelo de juego de seguridad de Stackelberg que resuelve la coordinación de múltiples recursos defensivos descentralizados. Este modelo asume la presencia de incertidumbre en las acciones efectuadas por cada recurso defensor y la ausencia de comunicación entre ellos.

En específico, el modelo de juego de seguridad de este trabajo consiste en resolver un número pequeño de problemas de programación lineal, que se pueden resolver mediante un esquema de generación de columnas. El subproblema de dicha generación de columnas corresponde a la resolución de un problema de decisión markoviana descentralizado. Estos problemas de decisión markoviana descentralizados son de difícil solución; sin embargo, es posible resolver estos subproblemas mediante heurísticas, dando como resultado un enfoque capaz de obtener soluciones subóptimas para el modelo de juego de seguridad.

Se presentan diversas heurísticas para la resolución de dicho subproblema, y se realiza un estudio para evaluar su uso dentro del esquema de generación de columnas. Este estudio consiste en la simulación de instancias de prueba aleatorias para evaluar el desempeño, tanto en el valor del resultado obtenido como en el tiempo de resolución de cada heurística. Se presenta una cota para el valor óptimo de un problema de decisión markoviana descentralizado que se obtiene al resolver un problema de optimización entero relacionado. Nuestros estudios computacionales muestran que esta cota es menor a un 10% del valor óptimo.

Se presentan además, variantes del enfoque de generación de columnas, buscando reducir los tiempos de solución sin sacrificar calidad de la respuesta. Estos enfoques también están basados en generación de columnas y otorgan una solución subóptima al problema planteado.

Con el objetivo de evaluar el comportamiento de los enfoques presentados, se recurre a la simulación de instancias aleatorias y una instancia inspirada en parte de la red de metro de Santiago. Además, con el objetivo de poder evaluar las soluciones subóptimas otorgadas por dichos enfoques, se desarrolla un método que permite obtener garantías para la solución del problema de generación de columnas. En específico, el algoritmo desarrollado permite resolver el problema de patrullar descentralizadamente una red conformada por 16 estaciones de metro, durante 12 períodos de tiempo y utilizando 6 recursos. Esta solución se obtiene, en promedio, en un tiempo de 400 [s] y con una garantía del 20%.

A mis padres y a mi abuela María Luisa.

Agradecimientos

Primero que todo, quiero agradecer a mi familia. En especial a mis padres, Fresia y Hugo, por todo el apoyo que me han entregado. Sin ellos, nada de lo que hoy he logrado sería posible. También agradecer a mis abuelos por todo el cariño. A mi abuela María Luisa, quien fue una segunda madre para mí. También a Carmen, quién siempre me ha apoyado en todas las etapas de mi vida.

Agradezco al profesor Fernando Ordoñez por guiar este trabajo, sin su valiosa contribución este trabajo no habría llegado a buen término. Muchas gracias por toda la buena disposición que tuvo conmigo, la confianza y las oportunidades que me brindó durante mi paso por el magíster. También agradecer su calidad humana, tanto en ámbitos académicos como fuera de ellos.

También agradezco a los miembros de la comisión, los profesores Denis Sauré y Richard Weber.

A Felipe Carrasco, a quién conocí durante mi estancia en el magíster, y he recibido su apoyo durante toda esta etapa.

A Linda Valdés por toda la ayuda entregada durante este proceso.

Agradezco a CONICYT por el financiamiento dado a este trabajo mediante la Beca CONICYT-PCHA/MagísterNacional/2015 - 22152053.

Finalmente, quiero agradecer a mis amigos, los cuales por diversos motivos no he listado aquí. Ellos saben el importante rol que han desempeñado a lo largo de mi vida.

Tabla de contenido

1. Introducción	1
1.1. Objetivos Generales de este Trabajo	2
1.2. Objetivos Específicos de este Trabajo	2
1.3. Estructura de la Tesis	3
2. Marco Teórico	4
2.1. Juegos de Seguridad de Stackelberg	4
2.2. Problema de Decisión Markoviano Descentralizado	8
2.3. Resolución de Problemas de Programación Lineal Mediante Generación de Columnas	12
2.4. Desigualdad de Jensen	14
3. Un Juego de Seguridad con Recursos Descentralizados	15
3.1. Antecedentes	15
3.1.1. Motivación de un Caso de Interés: Seguridad en una Red de Metro	16
3.2. Descripción del Modelo	18
3.3. Enfoque de Resolución para el Modelo	22
3.4. Formulación de Cotas	26
4. Desarrollo y Evaluación de Heurísticas para Dec-MDP	29
4.1. Heurísticas para Resolver Dec-MDP	29
4.1.1. Heurística Entropía Cruzada	29
4.1.2. Heurística Greedy	34
4.1.3. Heurística JESP	36
4.1.4. Heurística Solución a-priori	37
4.1.5. Heurística MILP y Cota para Problema Dec-MDP	37
4.2. Comparación de Heurísticas para Resolver Dec-MDP	41
4.2.1. Elección de Parámetros para Heurística Entropía Cruzada	45
4.2.2. Comparación de Heurísticas	46
5. Desarrollo de Experimentos Computacionales para Resolución del Modelo	52
5.1. Metodologías de Resolución para el Modelo de Juego de Seguridad	52
5.1.1. Enfoque Basado en Número Restringido de Cotas Superiores	54
5.1.2. Enfoque Basado en Cota Superior Probabilística	55
5.2. Experimentos Aleatorios	57
5.3. Ejemplo Representativo: Red de Metro	60
5.3.1. Visualización de Políticas	63

6. Conclusiones	65
Bibliografía	67
Anexos	72
Elección de Parámetros para Heurística Entropía Cruzada	72

Índice de tablas

4.1.	Parámetros que caracterizan a las instancias de cada conjunto.	42
4.2.	Valor de ψ para cada conjunto de instancias considerado.	45
4.3.	Valor de los parámetros requeridos por la heurística Entropía Cruzada para instancias correspondientes al conjunto N°1.	46
4.4.	Valor promedio e intervalo de confianza (método columna menos método fila) para instancias correspondientes al conjunto N°1.	49
4.5.	Tiempos de resolución de los diversos métodos utilizados para instancias correspondientes al conjunto N°1.	50
4.6.	Estadísticas del valor y tiempo de los métodos utilizados para los conjuntos de instancias N°2, N°3, N°4, N°5 y N°6.	51
5.1.	Valor de κ , λ , μ , α y β para cada conjunto de instancias considerado.	55
5.2.	Media y desviación estándar de la razón entre la solución heurística y su cota superior para cada conjunto de instancias considerado.	56
5.3.	Valor de ω para cada conjunto de instancias considerado.	57
5.4.	Tiempos de resolución al utilizar los enfoques #1, #2 y #3 para los conjuntos de instancias N°1, N°2, N°3, N°4 y N°5.	59
5.5.	Gap absoluto y relativo para los conjuntos de instancias N°1, N°2, N°3, N°4 y N°5.	60
5.6.	Tiempo de resolución mediante el enfoque #3 para instancias de la red de metro.	62
5.7.	Gap absoluto y relativo para instancias de la red de metro.	63
6.1.	Ajuste parámetro α para conjunto N°1.	73
6.2.	Ajuste parámetro I para conjunto N°1 (Tabla #1).	74
6.3.	Ajuste parámetro I para conjunto N°1 (Tabla #2).	75
6.4.	Ajuste parámetro I para conjunto N°1 (Tabla #3).	76
6.5.	Ajuste parámetro N para conjunto N°1 (Tabla #1).	77
6.6.	Ajuste parámetro N para conjunto N°1 (Tabla #2).	78
6.7.	Ajuste parámetro ρ para conjunto N°1.	79

Índice de figuras

3.1. Ejemplo de una red de metro.	17
3.2. Valor de la efectividad en función del número de recursos para diferentes valores de ξ	21
4.1. Ejemplo de partición del grafo para la utilización del enfoque basado en incorporación de solución a-priori.	37
4.2. Ejemplo de grafo para una instancia perteneciente al conjunto Nº1.	43
4.3. Ejemplo de grafo para una instancia perteneciente al conjunto Nº2.	43
4.4. Ejemplo de grafo para una instancia perteneciente al conjunto Nº3.	43
4.5. Ejemplo de grafo para una instancia perteneciente al conjunto Nº4.	44
4.6. Ejemplo de grafo para una instancia perteneciente al conjunto Nº5.	44
4.7. Ejemplo de grafo para una instancia perteneciente al conjunto Nº6.	44
4.8. Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº1.	47
4.9. Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº2.	47
4.10. Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº3.	48
4.11. Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº4.	48
4.12. Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº5.	48
4.13. Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº6.	49
5.1. Tiempos de resolución utilizando diversos enfoques para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.	58
5.2. Gap absoluto y relativo para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.	60
5.3. Grafo para la instancia de la red de metro.	61
5.4. Tiempo de resolución mediante el enfoque #3 para instancias de la red de metro.	62
5.5. Gap absoluto y relativo para instancias de la red de metro.	63
5.6. Visualización de la política empleada por un agente.	64
6.1. Ajuste parámetro N para conjunto Nº1.	76

Capítulo 1

Introducción

Uno de los principales desafíos que se debe afrontar para el otorgamiento de seguridad mediante la utilización de un sistema multiagente corresponde a la coordinación de los recursos defensivos que lo conforman. En específico, se requiere optimizar la utilización de dichos recursos, logrando la coordinación de ellos para que efectúen una labor de trabajo en equipo y maximicen la utilidad que recibe el sistema como un todo. Sin embargo; dicha coordinación no es trivial, ya que en los casos de aplicación en los que se utilizan estos sistemas existe incertidumbre sobre las acciones individuales que realiza cada uno de los recursos que lo conforman, y la comunicación entre ellos puede ser nula o limitada.

Un área que ha adquirido especial importancia dentro de los sistemas multiagentes corresponde a los juegos de seguridad, permitiendo la planificación exitosa de itinerarios para la utilización de múltiples recursos de vigilancia en la seguridad y protección de puertos marítimos, aeropuertos, complejos pesqueros, fronteras entre países limítrofes, contra la cacería ilegal de animales, entre otras aplicaciones [1], [2], [3], [4], [5], [6], [7].

El objetivo perseguido en un juego de seguridad es optimizar la utilización de un número limitado de recursos para la seguridad por parte del defensor; por ejemplo, determinando rutas de patrullaje aleatorias o zonas en las que realizar vigilancia. Esta utilización es optimizada tomando en cuenta la presencia de un adversario, el cual puede realizar vigilancia antes de ejecutar un ataque.

En este trabajo, se considera un modelo de juego de seguridad que resuelve la estrategia óptima de un conjunto de recursos defensores, los cuales se encuentran descentralizados, pero actúan de forma colaborativa. Este modelo asume las dificultades existentes para la coordinación de recursos, proporcionando así una formulación para el problema bajo dicho contexto. Dicho modelo es descrito para el caso específico de planificar itinerarios de patrullaje en una red de metro. Sin embargo, su aplicabilidad va más allá de este dominio en particular, siendo apto para diversas situaciones en donde se requiera el despliegue de un grupo de recursos de seguridad.

Además, se proponen nuevos enfoques para resolver el modelo mencionado, permitiendo abordar el problema de optimizar recursos descentralizados de seguridad que protegen alguna infraestructura. El desempeño de estos nuevos enfoques es medido mediante la resolución de instancias de prueba generadas de forma aleatoria. También, se diseñan instancias inspiradas en una red de metro, con el propósito de resolver una situación que se asemeje a un problema de seguridad de la vida real. Estos enfoques de solución heurísticos sólo logran obtener una solución subóptima para el problema presentado. Sin embargo, en este trabajo se desarrolla una forma de obtener cotas al problema de seguridad entregando una garantía a dichas soluciones, permitiendo así evaluar su calidad.

1.1. Objetivos Generales de este Trabajo

Los objetivos generales de este trabajo consisten en el desarrollo de algoritmos eficientes para la resolución de juegos de seguridad que consideran la coordinación de múltiples defensores descentralizados en presencia de un adversario estratégico. El problema abordado en este trabajo considera la presencia de incertidumbre en la ejecución de las acciones de los defensores. Para esto, dichas acciones se modelan mediante un proceso de decisión markoviana descentralizado. El modelo utilizado para representar el juego de seguridad de interés corresponde a un juego de Stackelberg. Este juego admite una descomposición mediante un esquema de generación de columnas, cuyo subproblema corresponde a un proceso de decisión markoviana descentralizado. Sin embargo, no se dispone de método para obtener solución óptima a dicho juego. En el presente trabajo se proponen diversos enfoques heurísticos, basados en generación de columnas, para la obtención de soluciones subóptimas de dicho problema. Se investigan también nuevas cotas al problema de decisión markoviana descentralizado considerado.

1.2. Objetivos Específicos de este Trabajo

Los objetivos específicos de este trabajo corresponden a desarrollar, implementar y evaluar métodos de solución eficientes, tanto para problemas de decisión markoviana descentralizados, como para algoritmos de descomposición eficientes basados en generación de columnas para resolver el juego de Stackelberg de seguridad con recursos descentralizados.

Otro de los objetivos específicos del presente trabajo consiste en formular una garantía para las soluciones obtenidas mediante los enfoques planteados, dado que dichos enfoques se basan en la utilización de heurísticas. Esto permite evaluar la calidad de las soluciones obtenidas para el modelo de juego de seguridad presentado en este trabajo. Se generarán instancias aleatorias para evaluar empíricamente la eficiencia de los algoritmos de generación de columnas para el juego de Stackelberg, los algoritmos heurísticos para resolver los problemas de decisión markoviana descentralizados, y las cotas desarrolladas. Adicionalmente se evaluará la utilidad de estos algoritmos en una instancia realista construida a partir de un problema de patrullaje en la red de metro de Santiago.

1.3. Estructura de la Tesis

La estructura de este trabajo es la siguiente: En el Capítulo 2 se presentan los fundamentos del modelo de juego de seguridad de Stackelberg utilizado, junto con las herramientas matemáticas utilizadas para su resolución. En el Capítulo 3 se detalla dicho modelo y se formula una cota superior teórica para su valor óptimo. En el Capítulo 4 se presentan heurísticas para resolver la problemática Dec-MDP, junto con un estudio, basado en simulación de instancias aleatorias, para determinar la de mejor desempeño que es entonces utilizada para resolver el juego de seguridad. En el Capítulo 5 se desarrollan enfoques de resolución basados en heurísticas para este modelo, los cuales son evaluados mediante simulación de instancias aleatorias, junto con una instancia inspirada en una red de metro real. Además, en este capítulo se obtienen garantías para la solución obtenida por dichos enfoques. El Capítulo 6 presenta las conclusiones de este trabajo. Finalmente, este trabajo contiene una sección de anexos.

Capítulo 2

Marco Teórico

2.1. Juegos de Seguridad de Stackelberg

Dentro de la disciplina de teoría de juegos, los juegos de Stackelberg [8] representan un problema de optimización binivel [9], en donde las decisiones de alto nivel son tomadas por un jugador que considera las variables óptimas de un problema de optimización anidado dentro del problema original. Este problema anidado resuelve la decisión óptima que debe tomar otro jugador como su mejor respuesta para las decisiones de alto nivel. En este tipo de problemas, el jugador que toma la decisión de alto nivel corresponde al líder, y mientras que el otro jugador es el seguidor.

En un juego de Stackelberg general (GSG, por sus siglas en inglés) se tiene K el conjunto de seguidores, I el conjunto de estrategias puras para el líder, y J el conjunto de estrategias puras para cada seguidor $k \in K$. Se denota por $\pi \in [0; 1]^{|K|}$ a la distribución de probabilidad sobre los $|K|$ seguidores. Se asume que el líder posee información perfecta sobre π . El *simplex* n -dimensional se denota por $S^n = \{\mathbf{x} \in [0; 1]^n : \sum_{i \in I} x_i = 1\}$. Una estrategia mixta para el líder consiste en un vector $\mathbf{x} \in S^{|I|}$ tal que para $i \in I$, x_i representa a la probabilidad con la que el líder utiliza la estrategia pura i . Similarmente, una estrategia mixta para el seguidor $k \in K$ corresponde a un vector $\mathbf{q}^k \in S^{|J|}$ tal que q_j^k es la probabilidad con la que el seguidor k emplea la estrategia pura $j \in J$. Los pagos para los jugadores dependen de las estrategias que empleen en conjunto, y vienen codificados en matrices de pago. Estas matrices se denotan por (R^k, C^k) , en donde $R^k \in \mathbb{R}^{|I| \times |J|}$ es la matriz de pagos para el líder cuando interactúa con un seguidor $k \in K$, y $C^k \in \mathbb{R}^{|I| \times |J|}$ es la matriz de pagos para el seguidor $k \in K$. En el resto de este trabajo se utiliza la palabra pago y utilidad como sinónimos.

El concepto de solución utilizado en este tipo de juegos corresponde al equilibrio fuerte de Stackelberg (SSE, por sus siglas en inglés) [10]. En un SSE el líder emplea una estrategia que maximiza su utilidad, anticipándose a que los seguidores ejecutarán su propia mejor respuesta a la estrategia del líder, para maximizar así su propia utilidad. El SSE asume que un seguidor romperá un empate en favor del líder eligiendo, entre todas las estrategias óptimas para el seguidor, la

estrategia que maximiza la utilidad para el líder. En la práctica, un SSE siempre existe, ya que es la solución del siguiente problema bilevel, que tiene funciones continuas sobre un conjunto acotado y factible. Un juego de Stackelberg puede ser modelado por el siguiente problema binivel:

$$\max_{(\mathbf{x}, \mathbf{q})} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \pi^k R_{ij}^k x_i q_j^k \quad (2.1)$$

s.a.

$$\mathbf{x} \in S^{|I|} \quad (2.2)$$

$$\mathbf{q}^k \in \arg \max_{\mathbf{r}^k \in S^{|J|}} \left\{ \sum_{i \in I} \sum_{j \in J} C_{ij}^k x_i r_j^k \right\}, \quad \forall k \in K. \quad (2.3)$$

El problema de primer nivel (problema para el líder) optimiza la recompensa esperada para el líder en la ecuación (2.1), mientras que la restricción (2.2) obliga a escoger una estrategia mixta. El problema de segundo nivel (problema para el seguidor) exige a cada seguidor $k \in K$ utilizar la estrategia $\mathbf{q}^k \in S^{|J|}$ que corresponda a la mejor respuesta para la estrategia adoptada por el líder, \mathbf{x} , para maximizar el pago del respectivo seguidor k . Para cada estrategia del líder \mathbf{x} y cada seguidor $k \in K$, la mejor respuesta para el problema que resuelve el respectivo seguidor viene dada por el vector $\mathbf{q}^k \in \{0; 1\}^{|J|}$, tal que $\sum_{j \in J} q_j^k = 1$.

El problema binivel presentado puede ser reformulado como un problema de programación lineal entera mixta (MILP, por sus siglas en inglés) [11]. Esta formulación es conocida como D2:

$$\max_{(\mathbf{x}, \mathbf{q}, \mathbf{s}, \mathbf{f})} \sum_{k \in K} \pi^k f^k \quad (2.4)$$

s.a.

$$\mathbf{x}^T \mathbf{1} = 1 \quad (2.5)$$

$$\mathbf{q}^k \in \{0; 1\}^{|J|}, \quad \forall k \in K \quad (2.6)$$

$$\mathbf{q}^{kT} \mathbf{1} = 1, \quad \forall k \in K \quad (2.7)$$

$$f^k \leq \sum_{i \in I} R_{ij}^k x_i + M(1 - q_j^k), \quad \forall k \in K, \quad \forall j \in J \quad (2.8)$$

$$0 \leq s^k - \sum_{i \in I} C_{ij}^k x_i \leq M(1 - q_j^k), \quad \forall k \in K, \quad \forall j \in J. \quad (2.9)$$

La ecuación (2.5) fuerza al líder a elegir una estrategia mixta, mientras que (2.6) exige que cada uno de los seguidores utilice una estrategia pura. En las ecuaciones (2.8) y (2.9), M es una constante positiva que tiene relación con valor del pago más alto. En la ecuación (2.8), f^k representa una cota para la utilidad del líder cuando interactúa con un seguidor de tipo $k \in K$. Esta cota se vuelve ajustada para la estrategia $j \in J$ utilizada por el seguidor. En la ecuación (2.9), s^k es una cota para la utilidad esperada del seguidor tipo $k \in K$. Esta cota se vuelve ajustada para la estrategia que corresponde a la mejor respuesta del seguidor respectivo. Las ecuaciones (2.8) y (2.9) en conjunto aseguran que la estrategia para el líder y las estrategias para cada uno de los seguidores mutuamente corresponden mejores respuestas. La función objetivo maximiza la utilidad esperada para el líder.

Se conoce como juegos de seguridad de Stakelberg (SSGs, por sus siglas en inglés) [12] aquellos juegos de Stackelberg en donde las estrategias para el líder consisten en la utilización de un número limitado de recursos para proteger una serie de objetivos, y las estrategias para los seguidores corresponden a atacar uno de estos objetivos. En este contexto, se refiere al líder como el defensor, mientras que los seguidores son referidos como atacantes. El conjunto de estrategias para los atacantes, J , consiste en n objetivos que pueden ser atacados, disponiendo de un conjunto de $m < n$ recursos que pueden ser utilizados para la protección de hasta m objetivos. El conjunto de estrategias puras para el defensor, I , está conformado por todos los $\sum_{i=1}^m \binom{n}{i}$ subconjuntos de a lo más m objetivos de J que el defensor puede proteger de forma simultánea. En los juegos de seguridad el defensor primero emplea una estrategia, mientras que el atacante observa dicha estrategia antes de adoptar una respuesta. Así, al utilizar el modelo de Stackelberg (líder y seguidor) como base para los juegos de seguridad, permite modelar la etapa de vigilancia del atacante previa a la realización de su ataque [12], [4], [13].

En este tipo de juego, los pagos sólo dependen del éxito o fracaso de los ataques sobre el conjunto de objetivos. Así, se denota por $D^k(j|c)$ al pago para el defensor cuando se enfrenta a un atacante de tipo $k \in K$ en un objetivo protegido $j \in J$, y por $D^k(j|u)$ al pago para el defensor cuando se enfrenta a un atacante de tipo $k \in K$ en un objetivo desprotegido $j \in J$. De igual forma, el pago para un atacante de tipo k cuando realiza un ataque sobre un objetivo desprotegido $j \in J$ se denota por $A^k(j|u)$, mientras que el pago para el atacante cuando ataca un objetivo protegido $j \in J$ se denota por $A^k(j|c)$. Los pagos en un juego de seguridad de Stackelberg se relacionan con los pagos en un juego de Stackelberg general mediante las ecuaciones (2.10) y (2.11). Así, el pago para el defensor corresponde a una recompensa o penalidad dependiendo de si la estrategia $i \in I$ es tal que se utiliza un recursos para proteger el objetivo $j \in J$; y similarmente, el pago para cada atacante $k \in K$ es una recompensa si la estrategia $i \in I$ no utiliza un recurso para proteger $j \in J$, o una penalidad si es que lo utiliza.

$$R_{ij}^k = \begin{cases} D^k(j|c) & \text{si } j \in i \\ D^k(j|u) & \text{si } j \notin i. \end{cases} \quad (2.10)$$

$$C_{ij}^k = \begin{cases} A^k(j|c) & \text{si } j \in i \\ A^k(j|u) & \text{si } j \notin i. \end{cases} \quad (2.11)$$

La finalidad de un juego de seguridad de Stackelberg es obtener la estrategia mixta para el defensor que maximiza la utilidad para él, dada la estrategia para el atacante, el cual posee información completa de las estrategias del defensor. En otras palabras, el cometido en un juego de seguridad corresponde a la optimización del uso de recursos de seguridad limitados por parte del defensor; por ejemplo, mediante la determinación de puntos de vigilancia o la utilización de rutas de patrullajes aleatorias. Esta optimización toma en consideración la presencia de un atacante que posee la habilidad de observar la estrategia mixta empleada por el defensor; y luego, responder óptimamente a dicha estrategia [11], [12], [14]. Esta finalidad es equivalente a la obtención de un SSE.

Una representación compacta del espacio de estrategias para el defensor, cuyo tamaño crece exponencialmente, se logra mediante la introducción, para cada objetivo $j \in J$, de una variable de

cobertura c_j [12]. Esta variable representa la frecuencia de protección en cada uno de los objetivos, y satisface que $c_j = \sum_{i \in \{I|j \in i\}} x_i$; es decir, la frecuencia con que un objetivo es protegido puede ser expresada como la suma de las probabilidades con que se utilizan las estrategias que protegen dicho objetivo. Así, una formulación binivel para el juego de seguridad de Stackelberg es la siguiente:

$$\max_{(\mathbf{c}, \mathbf{q})} \sum_{k \in K} \pi^k q_j^k \{c_j D^k(j|c) + (1 - c_j) D^k(j|u)\} \quad (2.12)$$

s.a.

$$\mathbf{c} \in [0; 1]^{|J|} \quad (2.13)$$

$$\mathbf{c}^T \mathbf{1} \leq m \quad (2.14)$$

$$q^k = \arg \max_{\mathbf{r}^k \in S^{|J|}} \left\{ \sum_{i \in I} \sum_{j \in J} r_j^k (c_j A^k(j|c) + (1 - c_j) A^k(j|u)) \right\}, \quad \forall k \in K. \quad (2.15)$$

La ecuación (2.12) maximiza la utilidad esperada para el defensor. Las ecuaciones (2.13) y (2.14) exigen que la estrategia de cobertura empleada por el defensor, \mathbf{c} represente una distribución de probabilidad sobre los objetivos y que su valor total de cobertura inducida esté acotado por el número de recursos disponibles. Al igual que antes, la ecuación (2.15) obliga a cada atacante $k \in K$ utilizar una estrategia q^k que maximice su propia utilidad, tomando en cuenta la estrategia de cobertura \mathbf{c} elegida por el defensor.

En los juegos de seguridad, la estrategia óptima para el defensor viene dada por la utilización de una estrategia mixta; sin embargo, para el caso del atacante sólo es necesario considerar estrategias puras [11]. Esto se debe a que, dada una estrategia mixta para el defensor, el atacante debe resolver un problema de optimización cuyas recompensas son lineales; y por lo tanto, si una estrategia mixta para el atacante es óptima, también lo son cada una de las estrategias puras en el soporte de dicha estrategia mixta. Así, no es necesario considerar estrategias mixtas para el atacante.

La formulación binivel presentada en las ecuaciones (2.12) a (2.15) puede ser reformulada como un MILP [12]. Esta formulación es conocida como **ERASER**:

$$\max_{(\mathbf{c}, \mathbf{q}, \mathbf{s}, \mathbf{f})} \sum_{k \in K} \pi^k f^k \quad (2.16)$$

s.a.

$$\sum_{j \in J} q_j^k = 1, \quad \forall k \in K \quad (2.17)$$

$$q_j^k \in \{0; 1\}, \quad \forall k \in K, \quad \forall j \in J \quad (2.18)$$

$$\sum_{j \in J} c_j \leq m \quad (2.19)$$

$$0 \leq c_j \leq 1, \quad \forall j \in J \quad (2.20)$$

$$f^k \leq D^k(j|c)c_j + D^k(j|u)(1 - c_j) + M(1 - q_j^k), \quad \forall k \in K, \quad \forall j \in J \quad (2.21)$$

$$0 \leq s^k - A^k(j|c)c_j - A^k(j|u)(1 - c_j) \leq M(1 - q_j^k), \quad \forall k \in K, \quad \forall j \in J \quad (2.22)$$

$$\mathbf{s}, \mathbf{f} \in \mathbb{R}^K. \quad (2.23)$$

Las ecuaciones (2.17) y (2.18) restringen a que un atacante de tipo $k \in K$ ataque a un solo objetivo $j \in J$. Las ecuaciones (2.19) y (2.20) aseguran de que las probabilidades de cobertura total en los objetivos no excedan el valor del número de recursos disponibles. En las ecuaciones (2.21) y (2.22), M corresponde a una constante positiva que tiene relación con el valor del pago más alto. Para un objetivo $j \in J$ atacado por un atacante de tipo $k \in K$, la ecuación (2.21) proporciona una cota ajustada para la utilidad esperada del defensor cuando éste se enfrenta a un atacante de tipo $k \in K$. Para cualquier otro objetivo, el lado derecho de (2.21) se vuelve grande, haciendo la restricción redundante. Similarmente, la ecuación (2.22) asegura que para cada atacante $k \in K$, s^k corresponda a una cota inferior para los pagos del atacante respectivo, y proporciona el pago óptimo de dicho atacante para el objetivo que ataca. La función objetivo maximiza la utilidad esperada para el defensor.

En [12] se muestra que cada vector factible de probabilidades de cobertura corresponde a una estrategia mixta del defensor; es decir, un despliegue de los recursos para proteger los objetivos; y que un vector de probabilidades de cobertura óptimo y un vector de estrategias óptimas para cada tipo de atacante corresponden a un equilibrio fuerte de Stackelberg del juego. Se han realizado trabajos previos en los que se compara el SSE contra otros tipos de equilibrio de Stackelberg, llegando a adoptarse comúnmente el SSE en juegos de seguridad [14], [15], [16], [17], [18], [12], [19], [4], [5], [20], [21].

Un método existente para la resolución de un juego de Stackelberg con un atacante corresponde al algoritmo de múltiples problemas lineales (Multiple-LP algorithm, por su nombre en inglés) [14]. Este método consiste en iterar sobre todas las posibles estrategias para el atacante, y para cada una de ellas calcular la estrategia óptima para el defensor. La solución óptima para el juego de seguridad corresponde a la estrategia para el defensor, entre todas las calculadas, cuya utilidad sea mayor. Este algoritmo es utilizado para resolver el juego presentado en este trabajo.

2.2. Problema de Decisión Markoviano Descentralizado

Los Procesos de Decisión Markoviana (MDPs, por sus siglas en inglés) [22] corresponden a modelos matemáticos que han sido exitosamente utilizados para la formalización de problemas de toma de decisiones secuenciales bajo la presencia de incertidumbre. Estos modelos permiten a un agente decidir cómo actuar en presencia de un ambiente estocástico con el objetivo de maximizar una cierta medida de desempeño. Se ha comprobado que este tipo de modelos son herramientas eficientes para resolver problemas de control monoagente, y han sido exitosamente aplicados a diversas situaciones de interés tales como robots móviles, manejo de inventario o problemas de flujo. Esto motiva a considerar la extensión formal de estos modelos para el control de sistemas multiagente colaborativos.

La extensión directa de un MDP al caso multiagente corresponde al Proceso de Decisión Markoviano Multiagente (MMDP, por sus siglas en inglés) [23]. Este modelo permite representar problemas de toma de decisiones secuenciales en escenarios multiagente cooperativos. Para esto se factoriza el espacio de acción de un MDP estándar en un espacio de acciones conjuntas; el cual se define como

un conjunto de acciones individuales, una por cada agente. Así, un MMDP puede ser visto como un gran MDP; ya que el conjunto de agentes puede ser representado como un solo ente, el cual busca resolver óptimamente un MDP en donde cada acción corresponde a una acción conjunta. Como consecuencia de esto, técnicas utilizadas para la resolución de MDP, tales como *policy iteration* o *value iteration*, pueden ser usadas para resolver un MMDP.

Este modelo representa un enfoque centralizado del problema, ya que hace el supuesto de que cada agente tiene completo conocimiento del estado global del sistema. La ventaja de este enfoque centralizado se traduce en que su resolución es más sencilla que para el caso descentralizado. Sin embargo, este supuesto es inapropiado para muchos de los sistemas multiagente de interés, ya que en ellos cada agentes sólo posee conocimiento parcial y diferente sobre el estado global del sistema.

Los Procesos de Decisión Markoviana Descentralizados (Dec-MDPs, por sus siglas en inglés) [24] corresponden a la extensión de MDP para el control descentralizado de sistemas multiagente. Este tipo de problemas está presente en situaciones en donde la realización de la acción de un agente podría depender en las acciones realizadas por los otros agentes. Uno de los casos más estudiados en donde se presenta este modelo considera el control de la operación de múltiples *rovers* para la exploración de planetas, como los utilizados por la NASA para la exploración de la superficie de Marte [25]. Otro caso de estudio que destaca corresponde al uso de UAVs (vehículos aéreos no tripulados, por sus siglas en inglés) para el reconocimiento y recolección de información en situaciones de combate [26].

Los Dec-MDPs contemplan múltiples agentes actuando de forma completamente descentralizada, cada uno de ellos eligiendo sus propias acciones basándose sólo en su propia incompleta y local observación del estado del sistema. Los agentes son cooperativos en el sentido de que existe una sola función objetivo del sistema que se intenta maximizar. Sin embargo, los agentes sólo pueden comunicarse entre sí durante la etapa de planificación, y no durante el horizonte temporal de ejecución. Por lo tanto, se debe contar con una política de acción conjunta (una política para cada agente) que no contemple intercambio de información entre los agentes. La definición formal de política conjunta se presenta más adelante en esta sección.

Formalmente un n -agente Dec-MDP se define como la tupla $\langle Ag, S, A, P, R, \Omega, O \rangle$, en donde:

- $Ag = \{1, \dots, n\}$ corresponde al conjunto de n agentes.
- S corresponde al conjunto finito de estados globales del sistema, con un estado inicial s^0 .
- $A = A_1 \times \dots \times A_n$ corresponde al conjunto finito de acciones conjuntas. A_i representa al conjunto de acciones que pueden ser utilizadas por el agente $i \in Ag$, mientras que se refiere a $a_i \in A_i$ como una acción local para dicho agente.
- $P : S \times A \times S \rightarrow \mathbb{R}$ corresponde a la función de transición. $P(s'|s, (a_1 \dots a_n))$ representa la probabilidad de que se presente el estado $s' \in S$ cuando se realiza la acción conjunta $(a_1 \dots a_n) \in A$ en el estado $s \in S$.
- $R : S \times A \times S \rightarrow \mathbb{R}$ corresponde a la función de utilidad. $R(s, (a_1 \dots a_n), s')$ representa la recompensa obtenida por el sistema al realizar la acción conjunta $(a_1 \dots a_n) \in A$ en el estado $s \in S$ y transicionando al estado $s' \in S$.

- $\Omega = \Omega_1 \times \dots \times \Omega_n$ corresponde al conjunto finito de observaciones conjuntas. Ω_i representa el conjunto observaciones del agente $i \in Ag$, mientras que se refiere a $o_i \in \Omega_i$ como una observación local para dicho agente.
- $O : S \times A \times S \times \Omega \rightarrow \mathbb{R}$ corresponde a la función de observación. $O(s, (a_1 \dots a_n), s', (o_1 \dots o_n))$ representa la probabilidad de que los agentes $\{1, \dots, n\}$ observen $\{o_1, \dots, o_n\}$ respectivamente, una vez ocurrida la secuencia $s, (a_1, \dots, a_n), s'$.

La formalización de un Dec-MDP incluye un horizonte temporal T , el cual especifica el número de períodos de toma de decisiones que son considerados.

Un Dec-MDP cumple la propiedad de observabilidad conjunta completa si la n -tupla de observaciones conjuntas realizada por los agentes determinan completamente el estado del sistema. Es decir; si $O(s, (a_1 \dots a_n), s', (o_1 \dots o_n)) > 0$, entonces $P(s' | (o_1 \dots o_n)) = 1$. Es decir, el estado global del sistema queda completamente determinado por el conjunto de observaciones locales de todos los agentes.

Se dice que el Dec-MDP es factorizado si cumple la propiedad de que el estado global del sistema se puede factorizar en $n+1$ componentes; es decir, $S = S_0 \times S_1 \times \dots \times S_n$. S_0 corresponde a características externas, las cuales son parte del estado global del sistema, pero no se ven afectadas por las acciones que los agentes realizan. S_i representa el conjunto de estados para el agente i , el cual corresponde a las características del estado global que un agente observa y puede influir. Esto permite la separación de las características del estado global que pertenecen a un agente en particular de aquellas que pertenecen al resto de ellos y a características externas. Esta separación es estricta, ya que ninguna característica del estado global del sistema puede pertenecer a más de un agente. Se refiere a $\hat{s}_i \in S_i \times S_0$ como el estado local para el agente i .

Un n -agente Dec-MDP factorizado se dice que es de transiciones independientes si existen funciones de transición locales P_0, \dots, P_n tales que:

$$P(s'_i | (s_0 \dots s_n), (a_1 \dots a_n), (s'_0 \dots s'_{i-1}, s'_{i+1} \dots s'_n)) = \begin{cases} P_0(s'_0 | s_0) & i = 0 \\ P_i(s'_i | \hat{s}_i, a_i, s'_0) & 1 \leq i \leq n \end{cases} \quad (2.24)$$

Es decir, el nuevo estado local de un agente en particular depende sólo en su estado local previo, la acción tomada por dicho agente, y las características externas al sistema presentes. Estas características sólo se ven afectadas por las características externas previas. Esto implica que:

$$P((s'_0 \dots s'_n) | (s_0 \dots s_n), (a_1 \dots a_n)) = P_0(s'_0 | s_0) \cdot \prod_{i=1}^n P_i(s'_i | \hat{s}_i, a_i, s'_0). \quad (2.25)$$

Un n -agente Dec-MDP factorizado se dice que es de observaciones independientes si existen O_1, \dots, O_n ; en donde O_i corresponde a la función de observación para el agente $i \in Ag$, tales que:

$$P(o_i | (s_0 \dots s_n), (a_1 \dots a_n), (s'_0 \dots s'_n), (o_1 \dots o_{i-1}, o_{i+1} \dots o_n)) = P(o_i | \hat{s}_i, a_i, \hat{s}'_i) \quad \forall o_i \in \Omega_i. \quad (2.26)$$

Es decir, la observación realizada por un agente en particular depende sólo en el estado local actual e inmediatamente siguiente de dicho agente, y de la acción que actualmente realiza.

Un n -agente Dec-MDP factorizado se dice que es observable de forma local y completa si:

$$\forall o_i \exists \hat{s}_i : P(\hat{s}_i | o_i) = 1. \quad (2.27)$$

Es decir, cada agente observa de forma completa su estado local en cada período. Aunque observabilidad de forma local y completa, y observabilidad independiente están relacionados, es factible que sólo se cumpla uno de dichos supuestos. Sin embargo, cuando ambos se cumplen los conjuntos Ω y O en la definición de un n -agente Dec-MDP factorizado son redundantes y pueden ser omitidos.

Una política local π_i (también llamada simplemente política) para el agente $i \in Ag$ corresponde a un mapeo entre secuencias de observaciones y acciones locales para dicho agente. Una política conjunta, $\{\pi_1, \dots, \pi_n\}$, es un conjunto de políticas, una para cada agente. Cada política local puede ser representada por una secuencia de reglas de decisión $\pi^i = \langle \sigma_i^0, \dots, \sigma_i^{T-1} \rangle$; en donde, σ_i^t corresponde al mapeo descrito para el período $t \in \{1, \dots, T-1\}$. Una regla de decisión descentralizada para el período τ corresponde a una n -tupla de reglas de decisión: $\sigma^\tau = (\sigma_1^\tau, \dots, \sigma_n^\tau)$, para $\tau \in \{1, \dots, T-1\}$. Las reglas de decisión pueden ser clasificadas como dependientes de la historia o markovianas.

Para el caso de ser dependiente de la historia, cada regla de decisión σ_i^τ mapea la historia local sobre estados y acciones de los τ períodos anteriores $h_i^\tau = \langle a_i^0, s_i^1, \dots, a_i^{\tau-1}, s_i^\tau \rangle$; hacia una acción local: $\sigma_i^\tau(h_i^\tau) = a_i^\tau$, para $\tau \in \{0, \dots, T-1\}$. En donde, $s_i^t \in S_i$ y $a_i^t \in A_i$ corresponden al estado y a la acción local, respectivamente, para el agente $i \in Ag$ en el período $t \in \{0, \dots, T-1\}$. Una secuencia de reglas de decisión dependientes de la historia define una política dependiente de la historia. En contraste, cada regla de decisión markoviana σ_i^τ mapea un estado local hacia una acción local: $\sigma_i^\tau(s_i^\tau) = a_i^\tau$, para $\tau \in \{0, \dots, T-1\}$. Una secuencia de reglas de decisiones markoviana define una política markoviana. Dado un cierto Dec-MPD, el número de sus políticas markovianas es exponencialmente menor que el número de sus políticas dependientes de la historia.

El objetivo de un Dec-MDP es encontrar una política descentralizada conjunta que maximice el valor esperado de la función de utilidad acumulada sobre el horizonte temporal T del problema. Esta función de utilidad viene dada por:

$$V^\pi(s_0) = \mathbb{E} \left[\sum_{t=0}^{T-1} R(s^t, a^t, s^{t+1}) | s^0, \pi \right]; \quad (2.28)$$

en donde, $s^t \in S$ y $a^t \in A$ corresponden al estado global del sistema y a la acción conjunta, respectivamente, en el período $t \in \{0, \dots, T-1\}$; y π representa a la política descentralizada conjunta utilizada.

Así, la política descentralizada conjunta óptima corresponde a:

$$\pi^* = \arg \max_{\pi} \left\{ \mathbb{E} \left[\sum_{t=0}^{T-1} R(s_t, a_t, s_{t+1}) | s_0, \pi \right] \right\}. \quad (2.29)$$

Una propiedad importante para los Dec-MDPs de transiciones y observaciones independientes, presentada en [27], corresponde a que sus políticas óptimas locales para cada agente sólo dependen del estado local de cada uno de ellos, y no en su historia. Así, una política markoviana descentralizada otorga el desempeño óptimo para un Dec-MDP de transiciones y observaciones independientes.

Este tipo de problemas es difícil de resolver. Se ha demostrado que la complejidad para resolver óptimamente un Dec-MDP es NEXP-completo, incluso en situaciones en donde se considera sólo dos agentes [28]. Esto contrasta con la complejidad para el mejor caso en MDP de P-completo [29].

2.3. Resolución de Problemas de Programación Lineal Mediante Generación de Columnas

En teoría de optimización lineal, un método para resolver problemas de programación lineal que poseen una gran cantidad de variables corresponde a generación de columnas (CG, por sus siglas en inglés) [30]. Sea el siguiente problema en forma estándar:

$$\min_{(\mathbf{x})} \mathbf{c}'\mathbf{x} \quad (2.30)$$

s.a.

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (2.31)$$

$$\mathbf{x} \geq 0; \quad (2.32)$$

en donde, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ y $\mathbf{A} \in \mathbb{R}^{m \times n}$. Se asume que la matriz \mathbf{A} está conformada por filas linealmente independientes entre sí; y que su número de columnas es inmensamente grande, haciendo imposible generarlas y almacenar la matriz completa en la memoria de una computadora. La experiencia con este tipo de problemas grandes sugiere que, usualmente, muchas de las columnas nunca logran entrar a la base (algoritmo Simplex [31]); y por lo tanto, es posible considerar no generar dichas columnas inservibles. Esta idea se alinea con el algoritmo Simplex, el cual en cada una de sus iteraciones, sólo requiere las columnas que forman la base actual y la columna que entra a la base. El enfoque de CG genera una columna de la matriz \mathbf{A} sólo una vez que ha determinado que puede entrar convenientemente a la base. Para esto se requiere de un método para descubrir variables x_i con costo reducido negativo, el cual corresponde a:

$$\bar{c}_i = c_i - \mathbf{c}'_{\mathbf{B}} \mathbf{B}^{-1} \mathbf{A}_i; \quad (2.33)$$

en donde, c_i corresponde al costo de la variable que entra a la base, $\mathbf{c}_{\mathbf{B}}$ al costo de las variables básicas actuales, \mathbf{B} a la matriz conformada por columnas básicas, y \mathbf{A}_i a la columna de la matriz \mathbf{A} correspondiente a la variable que entra a la base.

Una forma de descubrir dichas variables consiste en la resolución del siguiente problema:

$$\min_{(i)} \bar{c}_i, \quad (2.34)$$

en donde la minimización es sobre todo i . En muchos casos, el problema de optimización (2.34) posee una estructura especial, permitiendo la eficiente obtención del mínimo \bar{c}_i sin tener que calcular el valor de cada \bar{c}_i . Si el mínimo en este problema de optimización es mayor o igual que cero, todos los costos reducidos son no negativos y se tiene una solución óptima para el problema de programación lineal original. En el caso de que dicho mínimo sea negativo, la variable x_i correspondiente al índice minimizador i posee costo reducido negativo, y su respectiva columna \mathbf{A}_i puede entrar a la base.

La clave para el enfoque presentado radica en la capacidad de resolver el problema de optimización (2.34) de forma eficiente. Para algunos casos este problema presenta una estructura particular que hace sencilla su resolución; sin embargo, en el caso general existen problemas que no poseen la estructura deseada, imposibilitando la aplicación de la metodología descrita.

En la metodología presentada, las columnas que salen de la base son completamente descartadas. En una variante alternativa a esta metodología, el algoritmo retiene todas o algunas de las columnas que han sido generadas previamente, y procede considerando los problemas de programación lineal conformados sólo por dichas columnas.

Este algoritmo consiste en una secuencia de iteraciones de un problema maestro. En el comienzo de cada una de estas iteraciones, se tiene una solución básica factible para el problema original, y su respectiva matriz básica. Luego, se busca una variable cuyo costo reducido sea negativo. Una alternativa para dicha búsqueda corresponde a resolver la minimización de \bar{c}_i sobre el conjunto de índices i . Si ninguna variable de costo reducido negativo es encontrada, el algoritmo termina. En el caso que se encuentre una variable x_j tal que su respectivo costo reducido satisfaga $\bar{c}_j < 0$, su columna respectiva pasa a formar parte del conjunto de columnas del problema maestro. Este conjunto está formado por la colección de columnas $\mathbf{A}_i, i \in I$; el cual contiene todas las columnas básicas actuales, la columna recientemente anexada, \mathbf{A}_j , y posiblemente otras columnas. Se define el problema maestro como:

$$\min_{(\mathbf{x})} \sum_{i \in I} c_i x_i \quad (2.35)$$

s.a.

$$\sum_{i \in I} \mathbf{A}_i x_i = \mathbf{b} \quad (2.36)$$

$$\mathbf{x} \geq 0. \quad (2.37)$$

Las variables básicas que conforman una solución básica factible actual para el problema original están asociadas a columnas que han sido almacenadas en el problema maestro. Por lo tanto, se tiene una solución básica factible para el problema maestro, la cual puede ser utilizada como punto de partida para su resolución. Para esto, se realizan las iteraciones del algoritmo Simplex

necesarias para que el problema maestro sea optimizado. Luego, se procede con la siguiente iteración del problema maestro.

El método de generación de columnas descrito corresponde a un caso especial del algoritmo Simplex revisado; en donde, se utiliza reglas especiales para elegir la variable que entra a la base, las que le dan prioridad a las variables $x_i, i \in I$. Sólo una vez que los costos reducidos para dichas variables son todos no negativos, lo que ocurre cuando se obtiene la solución óptima para el problema maestro, el algoritmo examina el costo reducido de las demás variables. Esto surge de la motivación de darle prioridad a variables cuyas respectivas columnas ya han sido generadas, o a variables que son más probables de tener un costo reducido negativo.

Existen algunas variantes para este método, dependiendo de la forma en la que el conjunto I es elegido en cada iteración:

1. En un extremo, el conjunto I sólo corresponde al conjunto de índices de las variables básicas actuales, junto con la variable que está entrando a la base. Una variable que sale de la base queda inmediatamente fuera del conjunto I . Ya que el problema restringido posee $n+1$ variables y m restricciones, este problema es resuelto en una sola iteración del algoritmo Simplex, lo que ocurre cuando la columna \mathbf{A}_j entra a la base.
2. En el otro extremo, I corresponde al conjunto de índices de todas las variables que han entrado a la base en algún momento. Así, ninguna variable es descartada del conjunto I una vez que ha entrado a él. Para el caso de que se requiera de un gran número de iteraciones del problema maestro, esta variante puede causar problemas, ya que el conjunto I crecerá en cada una de estas iteraciones.
3. Finalmente, existe opciones intermedias, en las que el conjunto I es permitido tener un tamaño moderado, descartando aquellas variables que han salido de la base del problema y no han vuelto a entrar en un número predeterminado de iteraciones del problema maestro.

2.4. Desigualdad de Jensen

En matemáticas, la desigualdad de Jensen [32] relaciona el valor de una función convexa evaluada en una integral, con el valor de esa integral evaluada en dicha función convexa.

Sea (Ω, A, μ) un espacio medible tal que $\mu(\Omega) = 1$, g una función real μ -integrable y φ una función real convexa; entonces, la desigualdad de Jensen corresponde a:

$$\varphi\left(\int_{\Omega} g \, d\mu\right) \leq \int_{\Omega} \varphi \circ g \, d\mu. \quad (2.38)$$

Dada la generalidad de esta desigualdad, es posible de ser aplicada en múltiples contextos. En particular, para el caso de probabilidades, esta desigualdad puede ser reescrita como:

$$\varphi(\mathbb{E}\{X\}) \leq \mathbb{E}\{\varphi(X)\}. \quad (2.39)$$

Capítulo 3

Un Juego de Seguridad con Recursos Descentralizados

3.1. Antecedentes

Muchos de los trabajos que utilizan los juegos de seguridad presentados en el Capítulo 2 no consideran un importante desafío para el conjunto de recursos defensores. Mientras que la utilización de múltiples agentes defensores es optimizada, no se toma en cuenta el efecto de la presencia de incertidumbre en la coordinación de dichos agentes. En muchos de los contextos en que se requiere seguridad, la coordinación de equipos conformados por múltiples recursos defensores de diversos tipos (ej: patrullajes aéreos, recursos motorizados y caninos, entre otros) es necesaria para la efectividad del defensor. Sin embargo, esta coordinación multiagente presenta ciertas dificultades debido a los siguientes factores. Primero, al requerir que múltiples agentes coordinen sus acciones bajo la presencia de incertidumbre, retrasos surgidos a partir de situaciones inesperadas pueden producir la descoordinación de ciertos recursos, impidiendo una correcta acción conjunta. Segundo, algunos recursos pueden abandonar el sistema inesperadamente, requiriendo que otros suplan sus funciones. Tercero, los agentes pueden tener que actuar sin la posibilidad de comunicación entre ellos; por ejemplo, en algunas situaciones la comunicación es intencionalmente inhabilitada.

El modelo de juegos de seguridad que se describe en el presente capítulo, el cual es presentado en [33], resuelve el desafío planteado anteriormente, permitiendo optimizar la utilización del equipo defensor, a la vez que considera la existencia de incertidumbre en la coordinación de múltiples recursos. Para esto, se combinan dos áreas de estudio en sistemas multiagente: juegos de seguridad y cooperación multiagente bajo incertidumbre.

3.1.1. Motivación de un Caso de Interés: Seguridad en una Red de Metro

El modelo descrito en el presente capítulo puede ser aplicado a muchos de los casos de estudio en el ámbito de juegos de seguridad, incluyendo seguridad en vuelos, puertos y trenes [34]. Sin embargo, a modo de ilustrar un ejemplo en concreto se considera el caso de una red de metro. Este dominio es utilizado para detallar el modelo aquí descrito. Los desafíos presentados anteriormente de coordinación del equipo defensor bajo la presencia de incertidumbre, interrupciones en la ejecución de la estrategia desempeñada por algún agente, y comunicación limitada no son exclusivos para este dominio, y pueden presentarse en otros casos de interés. En el resto de este trabajo se utiliza la palabra agente y recurso como sinónimos (un agente representa un recurso).

En una red de metro los recursos defensores realizan patrullaje de las estaciones mientras que el adversario realiza vigilancia, pudiendo tomar ventaja al utilizar la predecibilidad del defensor para planificar un ataque. Dado que se dispone de recursos limitados para realizar labores de patrullaje, resulta imposible para el defensor cubrir todas las estaciones en cada instante de tiempo. Es por esto que el defensor debe decidir inteligente cómo realizar el patrullaje de las estaciones que conforman la red de metro. Este dominio presenta la restricción que los recursos defensores deben viajar en los trenes del metro para desplazarse, lo que los limita en la secuencia de estaciones que pueden vigilar. Además, los agentes deben adherirse a los itinerarios diarios de los trenes para la planificación de la ruta que deben realizar. Algunos de los trabajos en juegos de seguridad enfocados la problemática de la red de metro incluyen la obtención de itinerarios de patrullaje aleatorios para la red de metro de Singapur [35] y patrullajes de seguridad para la inspección de evasión en el sistema de metro de Los Angeles [18].

En la Figura 3.1 se muestra un ejemplo de la red de metro. Cada uno de los círculos representa una estación, mientras que cada una de las líneas rectas que atraviesan las estaciones corresponde a una de las diferentes líneas de la red del metro. Así, este ejemplo está constituido por 11 estaciones y 3 líneas. Por ejemplo, una de las líneas está conformada por las estaciones $\{t_1, t_3, t_6, t_9, t_{11}\}$. Otra de las líneas corresponde a las estaciones $\{t_2, t_3, t_4, t_5\}$. Las estaciones tienen diferentes pagos; por ejemplo, alguna estación puede ser lugar de combinación entre más de una línea del sistema de metro, siendo más atractiva de recibir un ataque por parte del adversario. En la figura este tipo de estaciones corresponde a t_3 y t_9 , y son representadas mediante un círculo ennegrecido. Además, se muestran dos posibles patrullajes que un recurso defensor puede ejecutar, el patrullaje #1 en color azul y el #2 en rojo, siendo infactible que sólo un recurso visite todas las estaciones dado la limitación temporal. El camino que recorre el patrullaje #1 comienza en la estación t_2 , viaja a la estación t_3 , luego visita la estación t_6 , luego t_9 , y finalmente termina en la estación t_{11} .

Los recursos defensivos pueden emplear trabajo en equipo para patrullar ciertas locaciones clave, resultando en un despliegue ventajoso para combatir al adversario en comparación a un patrullaje no colaborativo. Los recursos pueden realizar múltiples patrullajes y coordinarse para visitar una estación objetivo simultáneamente. Por ejemplo, los patrullajes #1 y #2 en la Figura 3.1, podrían coordinarse para encontrarse en la estación t_9 . De esta forma, si el adversario observa a múltiples recursos coordinados patrullando una estación, tendrá que enfrentarlos a todos si decide atacar.

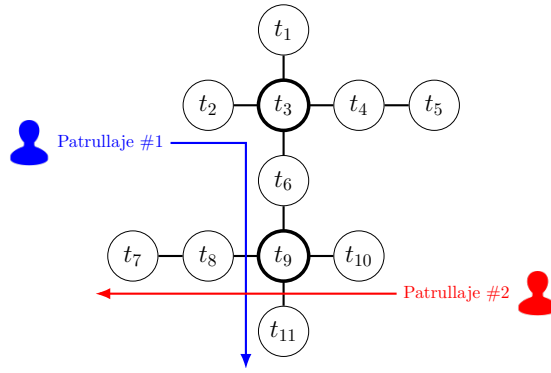


Figura 3.1: Ejemplo de una red de metro.

Dada la restricción de que los recursos deben viajar a través de una red de metro que se adhiere a un itinerario fijo diario para las salidas y llegadas de los trenes, se modela el tiempo como instantes discretos. Estos instantes de tiempo se basan en dicho itinerario. La utilización de un tiempo discreto permite que las acciones individuales de un recurso están basadas en la llegada y salida de los trenes, permitiendo a los recursos defensores llegar simultáneamente a una cierta estación.

En el caso del metro es posible identificar algunos factores que complican la realización del trabajo en equipo por parte de los recursos defensores. Primero, la incertidumbre existente en la ejecución de una estrategia por parte de un agente puede originar descoordinación entre los recursos. Por ejemplo, mientras los agentes se encuentran realizando un patrullaje, uno o más de ellos pueden verse obligados a desviar sus acciones de lo planificado debido a eventos imprevistos dada la existencia de incertidumbre. Dentro de estos eventos se encuentran el control de individuos sospechosos, el no lograr adherirse al itinerario de algún viaje en metro, entre otros; causando retrasos e incertidumbre en el patrullaje. El control de individuos conlleva a que el agente destine tiempo adicional para determinar el motivo o acción de estos individuos, teniendo que destinar un mayor tiempo en la ubicación en que se encuentra. Esto podría potencialmente causar que dicho agente pierda el próximo tren y descoordine las acciones de todos los recursos defensores. En la Figura 3.1 este tipo de incertidumbre puede causar que los patrullajes #1 y #2 visiten la estación t_9 en diferentes instantes, en vez de que lo hagan simultáneamente.

Segundo, en una red de metro a menudo hay comunicación limitada entre los recursos. Algunas de las razones para ello corresponden a que los trenes y las estaciones se encuentran bajo el nivel del suelo, el uso restringido de teléfonos móviles para evitar la detonación de bomba, o la utilización de radios por parte de los recursos defensores para no delatar información sobre sus acciones. Esto evita que los agentes estén constantemente comunicándose entre sí para determinar sus respectivas ubicaciones.

3.2. Descripción del Modelo

En la presente sección se detalla un modelo de juego de seguridad para el trabajo en equipo efectivo de múltiples recursos defensores descentralizados en presencia de incertidumbre. Esto corresponde a una adaptación del modelo de juego de seguridad descrito en el Capítulo 2 para la coordinación de múltiples recursos defensores. En particular, este modelo es aplicado al caso de la red de metro anteriormente descrito.

Este modelo permite enriquecer los juegos de seguridad mediante la inclusión de estrategias más complejas para el defensor, donde múltiples agentes deben coordinarse bajo incertidumbre para cumplir un mismo objetivo. Para esto, este modelo hace uso de los procesos de decisión markoviana descentralizados, introducidos en el Capítulo 2, en un componente específico del algoritmo empleado para la obtención de una estrategia mixta óptima para el defensor. Este componente involucra la problemática de un equipo multiagente, con incertidumbre sobre sus acciones, y sólo información local de los estados. Como se mencionó anteriormente, obtener la solución óptima para un Dec-MDP es complejo.

Sin embargo, no es posible aplicar directamente el modelo Dec-MDP para la resolución del juego de seguridad mencionado. Esto se debe a que en los juegos de seguridad, el defensor y el atacante tienen diferentes pagos asociados, lo cual no es posible de ser modelado mediante un Dec-MDP. Además, lo que se busca modelar son interacciones de teoría de juegos, en las que los pagos dependen de las estrategias utilizadas por el defensor y el atacante, siendo dicho modelo inapropiado para resolver estas interacciones. Con el objetivo de lograr obtener la estrategia mixta óptima para el defensor en el caso de presencia de incertidumbre, se descompone el problema en una componente de teoría de juego y una componente Dec-MDP (que se enfoca sólo en modelar la interacción entre los agentes defensores, y no requiere de la modelación de la interacción con el atacante ni considera sus diferentes pagos asociados).

Para modelar la problemática de interés se considera que tanto el atacante como el defensor disponen de un posible conjunto de estrategias puras. Para el caso del atacante las estrategias puras corresponden al conjunto de tuplas lugar-tiempo, $b = (t, \tau) \in B$. Cada una de estas tuplas se define con t siendo el lugar en donde se realiza el ataque y τ el instante de tiempo en el que éste se lleva a cabo. En el caso de la red de metro, los lugares corresponden a las estaciones de la red, mientras que el instante de tiempo corresponde a uno de los períodos discretos de tiempo en los que se define el problema. Las estrategias puras para el defensor, conformado por el grupo de agentes defensores, corresponden a visitar un conjunto de tuplas lugar-tiempo dados los recursos disponibles.

Este juego de seguridad presenta dos etapas de juego: el defensor emplea una estrategia mixta y a continuación el atacante responde eligiendo una tupla lugar-tiempo, basándose en la cobertura marginal otorgada por el defensor, en donde atacar; luego el juego termina.

Los pagos tanto para el defensor como para el atacante dependen de si la tupla lugar-tiempo está protegida o desprotegida por parte del defensor, acorde a la estrategia empleada por éste. Las acciones del defensor, junto con sus capacidades, determinan la efectividad de cobertura en estas tuplas, permitiendo una efectividad parcial. Cada tupla $b \in B$ tiene pagos asociados a ella tanto para el defensor como para el atacante. Se define por $U_d^c(b)$ el pago para el defensor si b es atacado y está protegido (100% de efectividad protectora), y $U_d^u(b)$ el pago para el defensor si b es atacado y está desprotegido (0% de efectividad protectora). La razón de tener pagos tanto para los lugares y tiempos se debe al hecho de que, en los dominios de interés, estos pagos son dependientes del tiempo. Por ejemplo, para el caso del sistema de metro, los pagos para una misma estación son mayores en el horario punta que en el horario valle. Estos pagos se ven influenciados por la cantidad de pasajeros que transitan en una estación a una determinada hora, ya que si el ataque es llevado a cabo en presencia de una gran cantidad de personas, éste resultará en una gran cantidad de víctimas en comparación a un ataque realizado en presencia de poco público.

De forma análoga se definen los pagos para el atacante, $U_a^c(b)$ y $U_a^u(b)$. Un supuesto común en los juegos de seguridad corresponde a $U_d^c(b) > U_d^u(b)$ y $U_a^c(b) < U_a^u(b)$. Es decir, cuando el defensor protege una cierta tupla b atacada, éste recibe una recompensa mayor, mientras que el atacante una menor, en comparación a la situación en donde no se protege b [36]. El modelo considerado permite un juego de suma no cero; es decir, la suma de los pagos para el atacante y defensor puede ser distinta de cero.

El equipo de agentes defensores se modela como un Dec-MDP factorizado que cumple los supuestos de transiciones independientes, observaciones independiente, y observabilidad de forma local y completa. Así, el modelo para los recursos puede ser representado mediante la tupla $\langle Ag, S, A, T, U \rangle$. En donde, $Ag = \{1, \dots, n\}$ representa el conjunto de n agentes o recursos defensivos. Para la problemática de interés presentada en este trabajo, las características externas son invariantes y pueden ser omitidas en la definición del estado global. Con esto, el conjunto de estados globales del sistema corresponde a $S = S_1 \times \dots \times S_n$. El estado local del agente $i \in Ag$ corresponde a la tupla $s_i = (t_i, \tau_i) \in S_i$, donde t_i corresponde al lugar en el que se encuentra el agente i (estación de metro) y τ_i es el tiempo en el que visita dicho lugar. El tiempo es considerado discreto, como se mencionó anteriormente, y se considera m períodos de decisión: $\{1, \dots, m\}$. $A = A_1 \times \dots \times A_n$ respresenta las acciones conjuntas. A_i corresponde al conjunto de acciones locales para el agente $i \in Ag$ y está conformado por las decisiones de qué lugar visitar en el siguiente período de tiempo. Estas decisiones corresponden a los posibles movimientos que puede hacer dicho agente dado el estado en el que se encuentra; los cuales pueden ser cambiarse a una estación que está directamente conectada a la de su ubicación actual, o quedarse en la estación en donde actualmente se encuentra. $T : S \times A \times S \rightarrow \mathbb{R}$ corresponde a la función de transición. Al cumplirse el supuesto de transiciones independientes, se tiene que $T(s, a, s') = \prod_{i \in Ag} T_i(s_i, a_i, s'_i)$; en donde $T_i(s_i, a_i, s'_i)$ es la función de transición para el agente i , introducida en la ecuación (2.24). Esta última función corresponde a la probabilidad de que el agente logre llegar a un cierto lugar en el siguiente período de tiempo dado que toma una determinada acción en su actual ubicación. Formalmente se modela el resultado de una acción local para un cierto agente de la siguiente manera: para cada acción local a_i en el estado local s_i existen dos estados locales, s'_i y s''_i , con probabilidad de transición no nula. s'_i es el estado conformado por el lugar pretendido en el siguiente período de tiempo; mientras

que s_i'' corresponde al mismo lugar que s_i , también en el siguiente período. Sin embargo, existe una diferencia considerable entre la tupla $\langle Ag, S, A, T, U \rangle$ descrita aquí y la tupla con que se define un Dec-MDP tradicional presentada en el Capítulo 2. Esta diferencia corresponde al elemento U , el cual representa la utilidad o recompensa de cada estado. Esta utilidad no sólo depende del estado o acciones del sistema, como lo es en un Dec-MDP típico, si no que depende de la interacción entre el defensor y el atacante.

Un itinerario de patrullaje simple, para cada uno de los recursos, consiste en una secuencia de órdenes. Cada una de ellas es de la forma: en el período τ , si el recurso se encuentra en la locación t , debe ejecutar la acción a . La acción de cada una de estas órdenes lleva al recurso defensor al lugar y tiempo de la siguiente orden. En la práctica, cada recurso coexiste con la presencia de incertidumbre, ya que cuando un recurso ejecuta una acción, puede resultar en que dicho recurso termine en el siguiente período de tiempo en un lugar diferente al pretendido. Este tipo de incertidumbre podría surgir, por ejemplo, a partir de los eventos imprevistos mencionados anteriormente.

La efectividad defensiva de un solo recurso que visita una cierta tupla lugar-tiempo se define como $\xi \in [0; 1]$. Este valor representa la probabilidad de que dicho recurso intercepte la estrategia de un atacante realizada en la tupla en donde se encuentra. El valor de ξ puede ser menor a 1, ya que visitar una cierta tupla lugar-tiempo no garantiza su protección por completo. Por ejemplo, si un agente visita para patrullar una cierta estación de metro en un período determinado, este recurso será capaz de brindar cierto nivel de efectividad; sin embargo, no puede garantizar que el adversario no realice un ataque. Dos o más recursos defensivos visitando una misma tupla lugar-tiempo logran una efectividad adicional. Dado un estado global $s \in S$ para los agentes, sea $\mathbf{eff}(s, b)$ la efectividad de los recursos en la tupla lugar-tiempo $b \in B$. El valor de la efectividad, $\mathbf{eff}(s, b)$, está también definido en el rango $[0; 1]$, con 0 significando nula cobertura y 1 completa protección para la tupla b . Se define la efectividad de k recursos visitando la misma tupla lugar-tiempo como $1 - (1 - \xi)^k$. Esta expresión corresponde a la probabilidad de impedir un ataque si cada uno de los recursos de forma independiente tiene probabilidad ξ de impedirlo. Así;

$$\mathbf{eff}(s, b) = 1 - (1 - \xi)^{\sum_{i \in Ag} I_{s_i=b}}; \quad (3.1)$$

en donde, $I_{s_i=b}$ corresponde a la función indicatriz que toma el valor 1 si $s_i = b$ y 0 en caso contrario. A medida que un mayor número de recursos visita la misma tupla lugar-tiempo, la efectividad aumenta, hasta un valor máximo de 1. Este aumento en el valor de la efectividad es decreciente al aumentar dicho número de recursos. Esto se debe a que la ecuación (3.1) corresponde a una función cóncava, hecho que será de importancia en el Capítulo 4. A modo de ejemplo, la Figura 3.2 muestra esta función para diversos valores de ξ . El aumento en la efectividad cuando una mayor cantidad de recursos visitan la misma tupla lugar-tiempo, $b \in B$, se traduce en que haya una mayor disuasión para el atacante, al observar dicha tupla y notar la presencia de múltiples recursos defensores, de efectuar ahí su ataque. Si es que el atacante observara un sólo agente, él podría elegir realizar el ataque descrito, evadiendo sólo un recurso. Sin embargo, si hay múltiples recursos defensivos, será más difícil para el atacante lograr su cometido, teniendo que elegir una tupla lugar-tiempo diferente. Aunque, en este trabajo se utiliza la función del valor de la efectividad descrita, el algoritmos para resolver el juego de seguridad presentado podría ser aplicado con otras funciones de efectividad, incluyendo el caso cuando los recursos tienen diferente efectividad individual. El único requisito

que debe cumplir esta función, dado algún estado global $s \in S$ y la tupla lugar-tiempo $b \in B$, es que su valor debe estar en el rango $[0; 1]$. Otras posibilidades incluyen el caso de agentes que proporcionan una efectividad mayor a 0 cuando se encuentran emparejados a algún tipo especial de recurso defensivo.

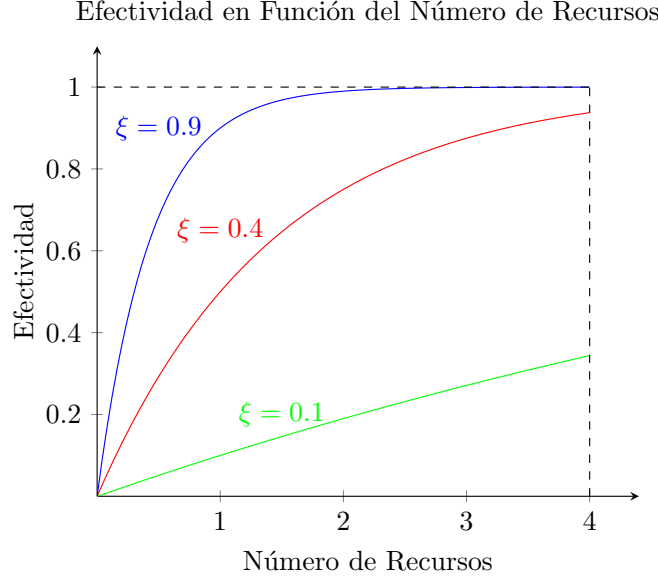


Figura 3.2: Valor de la efectividad en función del número de recursos para diferentes valores de ξ .

Se denota por π^j a la j -ésima estrategia pura del equipo defensor, también llamada política conjunta; y π^J al conjunto de todas las estrategias puras del defensor, siendo J el conjunto de índices correspondiente. Para el modelo presentado en este trabajo, cada una de estas estrategias puras se representa como una política conjunta de un Dec-MDP, en donde el conjunto de agentes Ag corresponde al grupo de recursos defensores. Por ejemplo, en el caso de haber dos recursos defensores, una política conjunta π^j incluye una política para el agente 1 (r_1) y una política para el agente 2 (r_2). La política de r_1 puede corresponder a: $\{((t_1, 0) : \text{Visitar } t_2), ((t_1, 1) : \text{Visitar } t_2), ((t_2, 1) : \text{Visitar } t_3)\}$, mientras que un ejemplo para la política de r_2 es: $\{((t_3, 0) : \text{Visitar } t_2), ((t_3, 1) : \text{Visitar } t_2), ((t_2, 1) : \text{Visitar } t_1)\}$. La política para r_1 es un mapeo entre el estado local de dicho agente y la acción que ahí le corresponde tomar. Si r_1 toma el estado $(t_1, 0)$, entonces la acción que debe tomar es visitar t_2 . Sin embargo, si r_1 toma el estado $(t_2, 1)$, entonces debe tomar la acción visitar t_3 . Al examinar la política conjunta, el recurso r_1 comienza en t_1 en el período 0, e intenta visitar t_2 y luego t_3 ; mientras que r_2 comienza en t_3 en el período 0, e intenta visitar t_2 y t_1 . El estado global del sistema en el período 0 corresponde a $\{(r_1 : (t_1, 0)), (r_2 : (t_3, 0))\}$, en donde r_1 se encuentra en t_1 y r_2 en t_3 .

Cada estrategia pura π^j induce una distribución de probabilidad sobre los estados globales del sistema. Se denota por $\mathbb{P}(s|\pi^j)$ a la probabilidad de que ocurra el estado global $s \in S$ dada la estrategia pura π^j . La efectividad esperada en la tupla lugar-tiempo $b \in B$, dada por una estrategia pura del defensor π^j , se denota por P_b^j y se define como:

$$P_b^j = \sum_{s \in S} \mathbb{P}(s|\pi^j) \text{eff}(s, b). \quad (3.2)$$

Dada una cierta estrategia pura del defensor π^j , y una estrategia pura del atacante correspondiente a una tupla lugar-tiempo $b \in B$, la utilidad esperada del defensor corresponde a:

$$U_d(b, \pi^j) = P_b^j U_d^c(b) + (1 - P_b^j) U_d^u(b). \quad (3.3)$$

Para el caso del atacante, su utilidad esperada se define de forma similar:

$$U_a(b, \pi^j) = P_b^j U_a^c(b) + (1 - P_b^j) U_a^u(b). \quad (3.4)$$

El defensor puede emplear una estrategia mixta \mathbf{x} , la cual corresponde a una distribución de probabilidad sobre el conjunto de estrategias puras. Generalmente la elección de una sola estrategia pura por parte del defensor, π^j , o una sola política conjunta, no corresponde a su estrategia óptima, debido a las variadas restricciones que limitan la cobertura sobre todas las posibles tuplas lugar-tiempo. Por ejemplo, una sola estrategia pura del defensor podría sólo permitir al equipo defensor visitar la mitad de todas las posibles tuplas lugar-tiempo. Si así fuese el caso, el atacante decidiría atacar a una de dichas tuplas que no está cubierta por el defensor. En esta situación, una estrategia mixta para el defensor que cubra todas las posibles tuplas lugar-tiempo constituye una mejor estrategia para él. Las utilidades esperadas para los jugadores, dada la utilización de estrategias mixtas, son naturalmente definidas como la esperanza de sus utilidades esperadas mediante el uso de estrategias puras. Formalmente, la utilidad esperada para el defensor dado el uso de la estrategia mixta \mathbf{x} por parte de él, y la estrategia pura $b \in B$ por parte del atacante, es $\sum_{j \in J} x_j U_d(b, \pi^j)$. Sea

$$c_b = \sum_{j \in J} x_j P_b^j \quad (3.5)$$

la cobertura marginal en $b \in B$ dada por la estrategia mixta \mathbf{x} ; y \mathbf{c} el vector de coberturas marginales sobre todas las tuplas lugar-tiempo. Esta cobertura marginal corresponde a la esperanza de la efectividad esperada. Se asume que el atacante observa la cobertura marginal del defensor para cada una de estas tuplas. Estas coberturas son definidas en función de la frecuencia del número de recursos en cada tupla. Es decir, el atacante toma en consideración que tan a menudo y con cuántos recursos cada una de ellas es visitada por el equipo defensor. La estrategia del atacante consiste en elegir una de estas tuplas y atacar. Una vez que sucede esto, se da por finalizado el juego.

Mediante las coberturas marginales es posible expresar la utilidad esperada para el defensor:

$$U_d(b, \mathbf{c}) = c_b U_d^c(b) + (1 - c_b) U_d^u(b). \quad (3.6)$$

De forma análoga, es posible expresar la utilidad esperada para el atacante como:

$$U_a(b, \mathbf{c}) = c_b U_a^c(b) + (1 - c_b) U_a^u(b). \quad (3.7)$$

3.3. Enfoque de Resolución para el Modelo

El objetivo de la formulación del juego descrito, la cual incluye las políticas conjuntas presentadas anteriormente como las estrategias puras del defensor, corresponde a obtener el equilibrio de

Stackelberg fuerte. En otras palabras, lo que se busca es encontrar la estrategia mixta óptima (cuyo valor esperado sea el mayor) para el defensor, considerando que existe un adversario estratégico que responde de forma óptima para dicha estrategia. Basándose en el modelo de teoría de juegos presentado en la sección anterior, la estrategia óptima para el defensor es obtenida mediante la resolución de un problema de programación lineal (LP, por sus siglas en inglés). Dado el número exponencial de estrategias puras para el defensor (o políticas conjuntas) que se requiere para resolver este LP, se utiliza un esquema de generación de columnas, presentado en el Capítulo 2, para generar de forma inteligente un subconjunto de dichas estrategias puras.

Para la resolución de este juego se hace uso del algoritmo de múltiples problemas lineales. En cada iteración de este algoritmo se asume que la mejor respuesta del atacante corresponde a una de las $|B|$ posibles estrategias puras; sea esta α , la cual consiste en una tupla lugar-tiempo $\alpha = (t, \tau)$.

$$\max_{(\mathbf{c}, \mathbf{x})} U_d(\alpha, \mathbf{c}) \quad (3.8)$$

s.a.

$$U_a(\alpha, \mathbf{c}) \geq U_a(b, \mathbf{c}), \quad \forall b \neq \alpha \quad (3.9)$$

$$c_b - \sum_{j \in J} P_b^j x_j \leq 0, \quad \forall b \in B \quad (3.10)$$

$$\sum_{j \in J} x_j = 1 \quad (3.11)$$

$$x_j \geq 0, \quad \forall j \in J \quad (3.12)$$

$$c_b \in [0; 1], \quad \forall b \in B. \quad (3.13)$$

El problema de programación lineal para α , presentado en las ecuaciones (3.8) a (3.13), calcula la estrategia mixta óptima \mathbf{x} para el defensor, dado que la mejor respuesta para el atacante corresponde a α . Luego, entre las $|B|$ soluciones obtenidas se elige la que posea mejor valor objetivo; es decir, la de mayor utilidad esperada para el defensor. Específicamente, la ecuación (3.9) obliga a que la mejor respuesta para el atacante sea α . En la ecuación (3.10), \mathbf{P}^j representa un vector columna, el cual contiene los valores de la efectividad esperada P_b^j para cada tupla lugar-tiempo $b \in B$, dada una cierta estrategia pura para el defensor π^j . Un ejemplo de estos vectores columna, tomado de [33], se presenta a continuación:

$$\mathbf{P} = \begin{matrix} & j_1 & j_2 & j_3 \\ \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{matrix} & \begin{bmatrix} 0.0 & 0.5 & 0.4 \\ 0.2 & 0.7 & 0.0 \\ 0.5 & 0.6 & 0.2 \\ 0.6 & 0.0 & 0.8 \end{bmatrix} \end{matrix}. \quad (3.14)$$

La columna $\mathbf{P}^{j_1} = \langle 0.0, 0.2, 0.5, 0.6 \rangle$ representa las efectividades esperadas $P_{b_i}^{j_1}$, dadas por la estrategia pura del defensor π^{j_1} , sobre cada tupla lugar-tiempo b_i ($i \in \{1, \dots, |B|\}$). Por ejemplo, la política π^{j_1} otorga a b_3 una efectividad de 0.5. De esta forma, la ecuación (3.10) obliga a que dada la probabilidad x_j de ejecutar una estrategia pura π^j , c_b corresponda a la cobertura marginal en

b.

El enfoque del algoritmo de múltiples problemas lineales resuelve un problema de programación lineal separado por cada una de las estrategias puras del atacante, denotada por α , correspondiente a las ecuaciones (3.8) a (3.13). Por ejemplo, el primer LP que es resultado asume que la mejor estrategia para el atacante corresponde al lugar t_1 en el período $\tau = 1$. Este algoritmo fija la mejor estrategia para el atacante, $\alpha = (t_1, 1)$; y luego optimiza la estrategia para el equipo defensor, restringido a que la mejor respuesta del atacante es α . Después, el algoritmo itera sobre el siguiente LP, el cual está conformado por una nueva estrategia para el atacante. Una vez que todos los LPs han sido resueltos, se comparan las estrategias del defensor para cada estrategia del atacante, y se elige aquella que presente la mayor utilidad esperada para el defensor.

En cada LP que se resuelve, el algoritmo requiere la mejor estrategia para el atacante, denotada por α , la cual está conformada por un lugar y un tiempo. Mientras que el resultado del algoritmo para cada LP corresponde a la estrategia para el defensor en contra de un atacante cuya mejor estrategia es α . Para determinar la estrategia óptima del defensor en contra del atacante, todas las estrategias puras del defensor deben ser enumeradas. Sin embargo, en el juego presentado existe un número exponencial de posibles estrategias puras para el defensor, las cuales corresponden a políticas conjuntas; y por lo tanto, un inmenso número de columnas que no son posibles de ser enumeradas en memoria computacional. Esto hace que no se pueda aplicar directamente el algoritmo de múltiples LPs. Para N estaciones, T períodos de tiempo, y R recursos defensivos, ya que se tiene la existencia de $(N^T)^R$ políticas diferentes.

Dado el crecimiento exponencial de las estrategias puras para el defensor en comparación al número de estaciones, períodos de tiempo, y recursos defensivos; se utiliza un enfoque de generación de columnas para resolver cada LP e inteligentemente generar un subconjunto de estrategias puras, junto con la obtención de la estrategia mixta óptima para el defensor. Se resuelve un LP utilizando el enfoque mencionado para cada una de las posibles tuplas lugar-tiempo para la estrategia del atacante, y luego se elige la solución que proporcione la mayor utilidad esperada para el defensor. El enfoque de generación de columnas está compuesto de dos componentes, el problema maestro y el subproblema. El problema maestro resuelve el LP dado un subconjunto de estrategias puras para el defensor. El subproblema proporciona la próxima mejor estrategia pura para el defensor con el fin de mejorar la solución obtenida por el problema maestro. Este subproblema es representado como un Dec-MDP para generar dicha estrategia pura.

El problema maestro consiste en un problema de programación lineal de la misma forma que el presentado en las ecuaciones (3.8) a (3.13), con la excepción de que en vez de poseer todas las estrategias puras, J ahora corresponde a un subconjunto de dichas estrategias. Para las estrategias puras que no pertenecen a J se asume que son empleadas con probabilidad nula, y sus respectivas columnas no necesitan ser representadas. Este nuevo LP es resuelto y se obtiene su solución óptima dual.

El propósito del subproblema corresponde a generar una estrategia pura para el defensor π^j y

añadir su columna correspondiente \mathbf{P}^j , la cual especifica las coberturas marginales, al problema maestro. A continuación se detalla cómo el problema de generar una buena estrategia pura puede ser traducido en resolver un Dec-MDP.

Para esto, con el objetivo de discernir si el hecho de incorporar una cierta estrategia pura π^j al problema maestro puede mejorar su solución óptima, se utiliza el concepto de costo reducido de una columna, presentado en la ecuación (2.33), el cual intuitivamente revela el potencial cambio en la función objetivo del problema maestro cuando una estrategia pura candidata π^j es añadida a él. Formalmente, el costo reducido \bar{f}_j asociado a la columna \mathbf{P}^j se define como:

$$\bar{f}_j = \sum_{b \in B} y_b \cdot P_b^j - z; \quad (3.15)$$

en donde, z corresponde a la variable dual de la ecuación (3.11) y $\{y_b\}$ son las variables duales de la familia de restricciones (3.10). Si $\bar{f}_j > 0$; entonces, añadir la estrategia pura π^j mejora el valor de la solución del problema maestro. En el caso de que $\bar{f}_j \leq 0$ para toda posible estrategia pura j , la solución actual del problema maestro es óptima para el LP completo.

Por lo tanto, el subproblema debe obtener la estrategia pura para el defensor π^j que maximice \bar{f}_j , y añadir su columna correspondiente al problema maestro en el caso de que $\bar{f}_j > 0$. Para el caso en que $\bar{f}_j \leq 0$, la generación de columnas termina y se obtiene la solución actual del problema maestro.

Para formular el problema de obtener la estrategia pura que maximiza el costo reducido se utiliza un Dec-MDP de transiciones independientes. La recompensa en cada estado es definida de tal forma que la recompensa esperada total es igual al costo reducido. Los estados y acciones para cada agente son definidos como se mencionó anteriormente. Es posible visualizar estos estados y acciones utilizando grafos de transición. Para cada recurso $r \in Ag$, el grafo de transición, $G_r = (N'_r, E'_r)$, contiene nodos de estado $s_r = (t, \tau) \in S_r$ para cada lugar y período de tiempo. Además, dicho grafo también contiene nodos de acción, $a_r \in A_r$, los cuales corresponden a las acciones que pueden ser empleadas en cada estado $s_r \in S_r$. Así, el conjunto de nodos N'_r está conformado por la unión de S_r y A_r . Existe un solo arco de acción entre un nodo de estado s_r y cada uno de los nodos de acción que corresponden a las posibles acciones que pueden ser efectuadas en s_r . Desde cada nodo de acción a_r proveniente de s_r hay múltiples arcos de posibilidad de salida hacia nodos de estado s'_r , los cuales corresponden a lugares en el siguiente período de tiempo. El conjunto de arcos E'_r queda conformado por la unión del conjunto de arcos de acción y de posibilidad. Cada uno de los arcos de posibilidad mencionados posee el atributo $T_r(s_r, a_r, s'_r)$, indicando la probabilidad de que ocurra la transición desde s_r , al emplear la acción a_r , hacia s'_r . En el contexto en el que se ha enfocado este trabajo, con la existencia de retrasos, cada nodo de acción posee dos arcos de posibilidad de salida; uno de ellos llegando al estado pretendido por la acción empleada; y el otro, a un estado diferente conformado por el mismo lugar que el inicial, pero en el siguiente período de tiempo.

El Dec-MDP de transiciones independientes mencionado en el párrafo anterior consiste en múltiples grafos de transición, uno para cada recurso $r \in Ag$, los cuales son denotados por G_r . Sin embargo, existe una función de recompensa conjunta $R(s)$. Esta función de recompensa conjunta

es dependiente de las variables duales del problema maestro, $\{y_b\}$, y de la efectividad $\mathbf{eff}(s, b)$ en la tupla lugar-tiempo $b \in B$ dada por los recursos cuyo estado global corresponde a $s \in S$, según se definió en la ecuación (3.1):

$$R(s) = \sum_{b \in B} y_b \cdot \mathbf{eff}(s, b). \quad (3.16)$$

Es necesario de un grafo de transición por cada uno de los recursos defensores, ya que cada uno de dichos recursos puede estar asociado a una estructura de grafo diferente o disponer de un conjunto de acciones particular.

A modo de ejemplo, para la función de recompensa conjunta $R(s)$ se utiliza el caso presentado en la Figura 3.1. El estado global corresponde a $s = \{(r_1 : (t_1, 0)), (r_2 : (t_3, 0))\}$; en donde, r_1 se encuentra en t_1 y r_2 en t_3 , para el período 0. Debido a que solamente hay dos tuplas lugar-tiempo en este estado global, sólo se requiere sumar sobre ellas, ya que para todas las demás la efectividad es nula ($\mathbf{eff}(s, b) = 0$). Si se tiene que $\xi = 0.6$, correspondiente a la efectividad defensiva de un solo recurso visitando a la tupla lugar-tiempo $b_1 = (t_1, 0)$ o $b_2 = (t_3, 0)$, entonces:

$$R(s) = \sum_{b \in B} y_b \cdot \mathbf{eff}(s, b) = y_{b_1} \cdot 0.6 + y_{b_2} \cdot 0.6. \quad (3.17)$$

La siguiente proposición, presentada en [33], provee una forma de obtener la estrategia pura para el defensor que maximiza el costo reducido del problema maestro mencionado anteriormente:

Proposición 1: sea π^j la solución óptima para el subproblema conformado por el Dec-MDP de transiciones independientes descrito y cuya función de recompensa está definida en (3.16). Entonces, π^j maximiza el costo reducido \bar{f}_j entre todas las estrategias puras para dicho problema.

3.4. Formulación de Cotas

El enfoque de generación de columnas presentado permite la obtención de una cota superior para el valor óptimo en el problema de programación lineal descrito por las ecuaciones (3.8) a (3.13), el cual corresponde a la formulación del juego de Stackelberg de interés en este trabajo. Más aún, esta cota superior es recalculada en cada iteración del algoritmo de generación de columnas. Para esto se reformula el problema de programación lineal mencionado.

La obtención de la estrategia mixta óptima para el defensor, \mathbf{x} , viene dada por la resolución del siguiente LP, en donde se asume que la mejor respuesta para el atacante corresponde a la estrategia

pura α . Sea (P) el siguiente problema:

$$\max_{(\mathbf{c}, \mathbf{x})} c_\alpha U_d^c(\alpha) + (1 - c_\alpha) U_d^u(\alpha) \quad (3.18)$$

s.a.

$$c_\alpha U_a^c(\alpha) + (1 - c_\alpha) U_a^u(\alpha) \geq c_\alpha U_a^c(b) + (1 - c_\alpha) U_a^u(b), \quad \forall b \neq \alpha \quad (3.19)$$

$$c_b - \sum_{j \in \mathcal{J}} P_b^j x_j \leq 0, \quad \forall b \in B \quad (3.20)$$

$$\sum_{j \in \mathcal{J}} x_j = 1 \quad (3.21)$$

$$x_j \geq 0, \quad \forall j \in J \quad (3.22)$$

$$c_b \in [0; 1], \quad \forall b \in B. \quad (3.23)$$

En donde, \mathbf{c} , \mathbf{x} , $U_d^c(b)$, $U_d^u(b)$, $U_a^c(b)$, $U_a^u(b)$, J , y P_b^j fueron definidos anteriormente.

Sea el problema (\bar{P}) , el cual es el mismo problema que (P) , excepto que en vez de contener todas las estrategias puras del defensor, sólo posee un subconjunto $\hat{J} \subseteq J$ de ellas. Sea v^* y v_{RM} los valores óptimos de (P) y \bar{P} , respectivamente. Claramente $v_{RM} \leq v^*$.

Sea (\bar{D}) el problema dual asociado al problema (\bar{P}) :

$$\min_{(\mathbf{y}, \mathbf{w}, \mathbf{z})} \sum_{b \neq \alpha} w_b (U_a^u(\alpha) - U_a^u(b)) + z + U_d^u(\alpha) \quad (3.24)$$

s.a.

$$w_b (U_a^c(b) - U_a^u(b)) + y_b \geq 0, \quad \forall b \neq \alpha \quad (3.25)$$

$$\sum_{b \neq \alpha} w_b (U_a^u(\alpha) - U_a^c(\alpha)) + y_\alpha \geq U_d^c(\alpha) - U_d^u(\alpha) \quad (3.26)$$

$$- \sum_{b \in B} y_b P_b^j + z \geq 0, \quad \forall j \in \hat{J} \quad (3.27)$$

$$y_b \geq 0, \quad \forall b \in B \quad (3.28)$$

$$w_b \geq 0, \quad \forall b \neq \alpha. \quad (3.29)$$

Sea (w, y, z) una solución para el problema (\bar{D}) . Se tiene que $z = \max_{j \in \hat{J}} \{ \sum_{b \in B} y_b P_b^j \}$. Se considera el problema de maximizar el costo reducido en (\bar{P}) :

$$p^* = \max_{j \in J} \left\{ \sum_{b \in B} y_b P_b^j \right\} - z. \quad (3.30)$$

Sea $\eta^* = \max_{j \in J} \{ \sum_{b \in B} y_b P_b^j \}$.

Se tiene que:

$$\begin{aligned}
v^* &\leq \sum_{b \neq \alpha} w_b (U_a^u(\alpha) - U_a^u(b)) + U_d^u(\alpha) + \max_{j \in J} \left\{ \sum_{b \in B} y_b P_b^j \right\} \\
&= \sum_{b \neq \alpha} w_b (U_a^u(\alpha) - U_a^u(b)) + U_d^u(\alpha) + p^* + z \\
&= v_{RM} + \eta^* - z
\end{aligned} \tag{3.31}$$

Por lo tanto, una cota inferior y superior para el valor óptimo de (\bar{P}) corresponden a:

$$v_{RM} \leq v^* \leq v_{RM} + \eta^* - z. \tag{3.32}$$

De esta forma, la ecuación (3.32) permite la obtención de una cota inferior y superior para el problema de programación lineal que resuelve el juego de Stackelberg para una cierta estrategia pura del atacante. La cota inferior, v_{RM} , corresponde a una solución factible para dicho problema; mientras que la cota superior, $v_{RM} + \eta^* - z$, representa una garantía para dicha solución. Estas cotas pueden ser recalculadas y actualizadas en cada iteración del algoritmo de generación de columnas. Sin embargo, obtener dicha cota superior requiere el valor de η^* , lo que es equivalente a resolver un Dec-MDP; como se mencionó en el Capítulo 2, esto es complejo. En el siguiente capítulo se presenta una manera computacionalmente eficiente de poder calcular esta cota superior en cada iteración del algoritmo de generación de columnas.

Capítulo 4

Desarrollo y Evaluación de Heurísticas para Dec-MDP

El modelo de juego de seguridad descrito en el Capítulo 3 presenta un espacio de políticas conjuntas muy grande. Con el objetivo de generar de forma progresiva el conjunto de políticas conjuntas se hace uso de un enfoque de resolución basado en generación de columnas, cuyo subproblema corresponde a un problema Dec-MDP. Así, la generación de una de estas políticas viene dada por la resolución de dicho problema. Sin embargo; como fue mencionado en el Capítulo 2, la resolución óptima para este tipo de problemas es compleja. Esto lleva que se considere su resolución mediante el uso de heurísticas, las cuales son descritas en el presente capítulo. La utilización de heurísticas en el algoritmo de generación de columnas no garantiza la obtención de la estrategia mixta óptima para el defensor. A pesar de esto, como se muestra en este capítulo, es posible obtener una garantía de la solución encontrada.

4.1. Heurísticas para Resolver Dec-MDP

4.1.1. Heurística Entropía Cruzada

En esta sección se describe una heurística que ha sido adaptada aquí para la resolución de un problema Dec-MDP. Su versión original está destinada para la resolución de procesos de decisión markoviana parcialmente observables descentralizados (Dec-POMDPs, por sus siglas en inglés) [37], [28], los cuales corresponden a una generalización de los Dec-MDPs presentados en el Capítulo 2. Esta heurística hace uso del método de entropía cruzada (CE, por sus siglas en inglés) [38], el cual es descrito a continuación.

Método de Entropía Cruzada para Problemas de Optimización

Uno de los enfoques para la obtención de soluciones aproximadas para problemas de optimización combinatorial corresponde al método de entropía cruzada. Este método ha sido provechosamente utilizado en varias aplicaciones, debido a su capacidad de encontrar, dentro de un inmenso espacio de búsqueda, soluciones cercanas al óptimo [39], [40], [41], [42]. En particular, se ha estudiado cómo este método puede ser adaptado para la obtención de buenas políticas en una cierta clase de MDPs [38], [42].

Debido a la naturaleza combinatorial del problema de control descentralizado, el método entropía cruzada resulta atractivo. Además, la aplicación del método para este tipo de problemas permite la búsqueda de soluciones en todo espacio de políticas conjuntas; a diferencia de otros métodos, como [43], [44], [45], [46]; que sólo realizan una búsqueda exhaustiva en un espacio restringido de soluciones.

El método de entropía cruzada puede ser utilizado en optimización cuando se presenta la problemática de encontrar un vector $x \in \mathcal{X}$, el cual maximiza cierta función de desempeño: $V : \mathcal{X} \rightarrow \mathbb{R}$. Es decir, cuando se tiene el siguiente problema:

$$x^* = \arg \max_{x \in \mathcal{X}} V(x). \quad (4.1)$$

El enfoque que se propone consiste en asociar un problema de estimación al problema de optimización presentado. Para esto se considera una distribución de probabilidad f_ξ sobre el conjunto factible \mathcal{X} , la cual se parametriza mediante el vector ξ . En específico, el método CE estima la probabilidad de que la función de desempeño para cierto vector x , tomado de la distribución f_ξ , sea mayor que un parámetro γ :

$$P_\xi(V(x) \geq \gamma) = \sum_{x \in \mathcal{X}} I(V(x), \gamma) f_\xi(x); \quad (4.2)$$

en donde, $I(V(x), \gamma)$ corresponde a la función indicatriz:

$$I(V(x), \gamma) = \begin{cases} 1 & : V(x) \geq \gamma \\ 0 & : V(x) < \gamma. \end{cases} \quad (4.3)$$

Sea γ^* el valor óptimo ($\gamma^* \equiv V(x^*)$). Considerando $\gamma = \gamma^*$ y una distribución uniforme para f_ξ en la ecuación (4.2) se obtiene el problema de optimización correspondiente a (4.1). Con esto, el método de entropía cruzada se traduce a un algoritmo iterativo que contempla dos etapas:

1. Generar un conjunto de N muestras, \mathbf{X} , tomado de la distribución f_ξ .
2. Seleccionar $\mathbf{X}_b \subset \mathbf{X}$, correspondiente a las N_b mejores muestras, y utilizarlo para actualizar el vector de parámetros ξ . El número de mejores muestras es caracterizado como una fracción $0 \leq \rho \leq 1$ de N .

Sea $\gamma^{(j)}$ el menor valor de la función de desempeño que se obtiene al evaluar los vectores correspondientes al conjunto \mathbf{X}_b en la j -ésima iteración del algoritmo. Es decir,

$$\gamma^{(j)} \equiv \min_{x \in \mathbf{X}_b} V(x). \quad (4.4)$$

Uno de los requerimientos del método CE corresponde a que esta cota inferior para la función de desempeño no decrezca durante las iteraciones del algoritmo: $\gamma^{j+1} \geq \gamma^j$. Esto conlleva a que el conjunto \mathbf{X}_b pueda tener menos de N_b muestras. Una vez obtenido el conjunto \mathbf{X}_b , este es utilizado para estimar el vector de parámetros ξ^{j+1} mediante máxima verosimilitud. El vector de nuevos parámetros puede ser suavizado a través de la su interpolación con el vector de parámetros de la iteración previa ξ^j y un factor $0 \leq \alpha \leq 1$:

$$\xi^{j+1} = \alpha \xi^{j+1} + (1 - \alpha) \xi^j. \quad (4.5)$$

Esta interpolación reduce la probabilidad de que alguna componente del vector de parámetros sea 0 ó 1 en las primeras iteraciones del algoritmo, lo que podría causar que el proceso converja hacia un óptimo local.

Usualmente la condición de término para el proceso consiste en detenerlo cuando $\gamma^{(j)}$ no ha mejorado durante un número predefinido de iteraciones. Por otro lado, un tiempo límite o número fijo de iteraciones también pueden ser utilizados como condición de término. Una vez alcanzada alguna de estas condiciones, la muestra x que presente mayor función de desempeño y encontrada durante todo el proceso es elegida como una aproximación de x^* .

Aplicación del Método para la Solución de MDPs

Una aplicación particular del método de entropía cruzada para optimización corresponde a su utilización para resolver MDPs. En [42] es utilizado para la solución del problema de camino mínimo, el cual se formula como un MDP cuyo valor esperado de la función de utilidad acumulada óptimo, para el horizonte temporal considerado, es estacionaria (la recompensa esperada de tomar una cierta acción en un determinado estado no depende del período correspondiente). De esta forma, la política óptima para esta clase de MDPs corresponde a un mapeo de estados a acciones $\pi^* : S \rightarrow A$, la cual puede ser representada como un vector de norma $|S|$. El objetivo buscado es encontrar una política que maximice la función de desempeño dada por la utilidad esperada total. Así, la ecuación (4.1) se reescribe como:

$$\pi^* = \arg \max_{\pi \in \theta} V(\pi); \quad (4.6)$$

en donde, θ corresponde al espacio de políticas para dicho problema. El método de entropía cruzada afronta este problema utilizando un vector de parámetros $\xi = \langle \xi_{S_1}, \dots, \xi_{S_{|S|}} \rangle$, en donde cada ξ_{S_i} , $i \in \{1, \dots, |S|\}$ representa una distribución de probabilidad sobre el conjunto de acciones. Mediante la utilización de estas distribuciones de probabilidad es posible obtener N trayectorias: comenzando en algún estado inicial las acciones son elegidas aleatoriamente según las distribuciones presentes

en ξ , hasta que un cierto estado final es alcanzado. Considerando las N_b trayectorias con mayor utilidad esperada total, \mathbf{X}_b , el vector de parámetros es actualizado de la siguiente forma:

$$P(a|s) = \frac{\sum_{x \in \mathbf{X}_b} I(x, s, a)}{\sum_{x \in \mathbf{X}_b} I(x, s)} \quad \forall a \in A, \quad \forall s \in S; \quad (4.7)$$

en donde, $I(x, s, a)$ corresponde a la función indicatriz asociada a la realización de la acción $a \in A$ en el estado $s \in S$ durante la trayectoria $x \in \mathbf{X}_b$. Mientras que $I(x, s)$ indica si el estado $s \in S$ fue visitado en la trayectoria $x \in \mathbf{X}_b$. Una vez que se ha actualizado el vector de parámetros ξ , es posible obtener un nuevo conjunto de N trayectorias hasta alcanzar la condición de término especificada para el proceso.

Aplicación del Método para el Caso de Dec-MDP

Una vez que se ha descrito cómo es posible aplicar el método de entropía cruzada para la resolución de MDPs, se puede realizar su extensión para resolver el caso de Dec-MDPs. Esta extensión se basa en el trabajo hecho en [47], en donde se presenta un método para resolución de Dec-POMDPs, adaptándolo para el caso de Dec-MDPs. Esta adaptación resulta sencilla, ya que los Dec-POMDPs corresponden a una generalización de los Dec-MDPs, como se mencionó anteriormente. Así, se llega a la formulación de un algoritmo llamado: búsqueda directa de políticas conjuntas mediante entropía cruzada (DiCE, por sus siglas en inglés).

En el caso de un Dec-MDP el espacio de soluciones factibles corresponde al espacio de políticas conjuntas deterministas Π . Para poder aplicar el método de entropía cruzada es necesario definir una distribución de probabilidad sobre dicho espacio, junto con una función de desempeño para evaluar políticas conjuntas tomadas de ella. Además, se requiere especificar cómo esta distribución es actualizada utilizando las muestras de políticas conjuntas que presentan mayor función de desempeño obtenidas en cada iteración del algoritmo.

Para la especificación de f_ξ , la distribución de probabilidad sobre políticas conjuntas puras antes mencionada y parametrizada mediante ξ , ésta puede ser representada como el producto de distribuciones de probabilidad sobre políticas puras individuales:

$$f_\xi(\pi) = \prod_{i=1}^n f_{\xi_i}(\pi_i). \quad (4.8)$$

En este caso, ξ_i especifica al vector de parámetros asociado al agente $i \in Ag$, mientras que el vector de parámetros para la distribución conjunta corresponde a $\xi = \langle \xi_1, \dots, \xi_n \rangle$. Con esto, el problema de representar una distribución de probabilidad sobre el espacio de políticas conjuntas se traduce a representar distribuciones de probabilidad sobre políticas individuales. El enfoque que se adopta para resolver este problema corresponde a describir la política individual de cada agente mediante una política estocástica. Para esto, se considera para cada uno de los agentes una distribución de probabilidad sobre el conjunto de acciones, para cada estado y período de

tiempo. Se requiere un parámetro ξ_i^{st} para cada agente y período de tiempo, el cual especifica la distribución sobre el conjunto de acciones que el agente $i \in Ag$ toma al encontrarse en el estado $s \in S_i$ durante el período $t \in T$:

$$\xi_i^{st}(a) = P(a|s, t) \quad \forall i \in Ag, \quad \forall s \in S_i, \quad \forall t \in T, \quad \forall a \in A_i. \quad (4.9)$$

Así, el vector de parámetros para el agente $i \in Ag$ se define como $\xi_i \equiv \langle \xi_i^{st} \rangle_{s \in S_i, t \in T}$; mientras que la probabilidad de que se obtenga una cierta política individual π_i para el agente i corresponde a:

$$f_{\xi_i}(\pi_i) = \prod_{s \in S_i, t \in T} \xi_i^{st}(\pi_i(s, t)). \quad (4.10)$$

A diferencia del caso presentado para MDPs, en donde se utiliza trayectorias para la actualización del vector de parámetros ξ , para el caso de Dec-MDPs la actualización de su correspondiente vector se realiza mediante la generación aleatoria de políticas conjuntas completas. Obtener una política conjunta mediante la distribución f_{ξ} se realiza de la siguiente manera. Para cada estado $s \in S_i$ y período de tiempo $t \in T$ en los que el agente $i \in Ag$ es factible de estar, una acción es generada de acuerdo a la distribución ξ_i^{st} . Con esto se obtiene una política determinista para dicho agente. Repitiendo este proceso para cada uno de los agentes en el conjunto Ag , se obtiene una política conjunta determinista, la cual puede ser evaluada mediante la función de desempeño correspondiente a la ecuación (2.28). Así, es posible obtener el conjunto \mathbf{X} de políticas conjuntas, y elegir el subconjunto \mathbf{X}_b de él.

Una vez que se dispone del conjunto \mathbf{X}_b obtenido en la j -ésima iteración del algoritmo, cuyas políticas conjuntas fueron generadas mediante la distribución $f_{\xi^{(j)}}$, éste es utilizado para obtener el vector de parámetros $\xi^{(j+1)}$ correspondientes a la siguiente iteración del algoritmo. Para esto, sea $I(\pi_i, s, t, a)$ la función indicatriz que indica si la acción $a \in A_i$ es tomada por la política π_i del agente $i \in Ag$ al encontrarse en el estado $s \in S_i$ durante el período $t \in T$. Con esto, la probabilidad de que el agente $i \in Ag$ tome una cierta acción $a \in A_i$ al encontrarse en el estado $s \in S_i$ durante el período $t \in T$ puede ser estimada nuevamente como:

$$\xi_i^{st(j+1)}(a) = \frac{1}{|\mathbf{X}_b|} \sum_{\pi \in \mathbf{X}_b} I(\pi_i, s, t, a) \quad \forall i \in Ag, \quad \forall s \in S_i, \quad \forall t \in T, \quad \forall a \in A_i; \quad (4.11)$$

en donde, $|\mathbf{X}_b|$ normaliza la distribución, ya que:

$$\sum_{a \in A_i} \sum_{\pi \in \mathbf{X}_b} I(\pi_i, s, t, a) = |\mathbf{X}_b| \quad \forall i \in Ag, \quad \forall s \in S_i, \quad \forall t \in T. \quad (4.12)$$

De esta forma es posible adquirir el vector de parámetros $\xi^{(j+1)}$, el cual previo a su utilización debe ser suavizado mediante la ecuación (4.5).

Sea I el número total de iteraciones del algoritmo, N el número de políticas conjuntas generadas en cada iteración, N_b en número de políticas conjuntas utilizadas para la actualización del vector ξ , y α el factor de suavización. El siguiente pseudocódigo presenta el algoritmo descrito.

Algorithm 1 Heurística Entropía Cruzada

Require: $[I, N, N_b, \alpha]$ **Ensure:** $[\pi_b]$ $V_b = -\infty.$ Initialize $\xi^{(0)}$ (uniform random).**for** $i = 0 : I$ **do** $\mathbf{X} = \emptyset.$ **for** $s = 0 : N$ **do**Draw $\pi \sim f_{\xi^{(i)}}$. $\mathbf{X} = \mathbf{X} \cup \{\pi\}.$ **if** $V(\pi) > V_b$ **then** $V_b = V(\pi).$ $\pi_b = \pi.$ **end if****end for**Select \mathbf{X}_b : the set of N_b best $\pi \in \mathbf{X}$.Compute $\xi^{(i+1)}$ according to (4.11). $\xi^{(i+1)} = \alpha \xi^{(i+1)} + (1 - \alpha) \xi^{(i)}.$ **end for**

4.1.2. Heurística Greedy

Es posible obtener una heurística, para el caso particular de los problemas Dec-MDP que se requieren resolver en este trabajo, mediante la utilización de un enfoque greedy. Sin embargo, este enfoque no puede ser utilizado para el caso de un Dec-MDP general. Esta heurística consiste en iterativamente resolver un problema MDP para cada uno de los agentes considerados en el conjunto Ag . Cada uno de estos MDPs difiere en su función de utilidad, la cual toma en consideración el efecto sobre el sistema de la solución otorgada por los agentes previamente resueltos.

El valor de esta función de utilidad queda determinado por las variables duales $\{y_b\}$ del problema maestro asociado al Dec-MDP de interés, presentadas en la ecuación 3.15, y las políticas óptimas individuales obtenidas en los MDPs correspondientes a recursos resueltos en iteraciones previas.

El algoritmo que conforma esta heurística requiere los grafos de transiciones, uno por cada recurso considerado, G_i ($\forall i \in Ag$) introducidos en el Capítulo 3. Este algoritmo retorna una política conjunta factible para el Dec-MDP en cuestión, π^j , la cual corresponde a la j -ésima estrategia pura para el defensor en el problema maestro asociado. El siguiente pseudocódigo presenta el algoritmo descrito.

Este algoritmo calcula una política óptima individual, π_i , para cada uno de los agentes $i \in Ag$. Este cálculo se realiza mediante la resolución estándar de un MDP para cada uno de los agentes, proceso que ocurre en la función “SolveSingleMDP”(·) del pseudocódigo presentado. En específico, el MDP para el agente $i \in Ag$ consiste en S_i , el conjunto de estados locales para el agente i ; A_i , el

Algorithm 2 Heurística Greedy

Require: $[y_b, G]$ **Ensure:** $[\pi^j]$ Initialize $\pi^j = \{\emptyset\}$.**for** $i = 1 : |Ag|$ **do** $\mu_i \leftarrow \text{ComputeUpdateReward}(\pi^j, y_b, G_i)$ $\pi_i \leftarrow \text{SolveSingleMDP}(\mu_i, G_i)$ $\pi^j \leftarrow \pi^j \cup \pi_i$ **end for**

conjunto de acciones que puede utilizar el recursos i ; $T_i(s_i, a_i, s'_i)$, la función de transición para el agente i , la cual indica la probabilidad de que dicho agente termine al siguiente período de tiempo en el estado local $s'_i \in S_i$ cuando toma la acción $a_i \in A_i$ en el estado local $s_i \in S_i$; y $R(s_i)$, la función de utilidad, la que representa la recompensa obtenida por el agente cuando visita el estado local $s_i \in S_i$.

Esta función de utilidad se especifica mediante el vector de utilidad, μ_i , para cada uno de los agentes considerados. Este vector corresponde a la utilidad que recibe dicho agente cuando visita cada uno de los estados del sistema. La forma de calcular dicho vector de utilidad requiere de las variables duales mencionadas, junto con las políticas individuales para los recursos que ya han sido calculadas previamente. Una vez que las políticas individuales para todos los recursos han sido generadas, se logra la obtención de la política conjunta buscada.

Sea un Dec-MDP constituido por los recursos Ag , cuya función de utilidad corresponde a 3.16. Para el MDP correspondiente a un cierto recurso $i \in Ag$ que resuelve el algoritmo descrito, su función de utilidad en un cierto estado $s_i \in S_i$, $\mu(s_i)$, corresponde a la contribución marginal que otorga dicho recurso al visitar el estado s_i a la función de utilidad conjunta, dada las políticas de los $i - 1$ recursos calculadas previamente: $\pi^j = \{\pi_1, \dots, \pi_{i-1}\}$. Gracias a la propiedad de transiciones independientes, dado π^j es posible calcular la probabilidad, $p_{s_i}(k)$, de que k de los $i - 1$ recursos hayan visitado el estado s_i . Con esto, $\mu(s_i) = \sum_{k=0}^{i-1} p_{s_i}(k)(\psi(k+1) - \psi(k))$; en donde, $\psi(k)$ es la efectividad que brindan k recursos defensivos. La actualización de dicha función de utilidad, para cada uno de los recursos que iterativamente resuelve el algoritmo, se actualiza mediante la función “ComputeUpdateReward”(·) presentada en el pseudocódigo. Esta función requiere de las políticas asociadas a los recursos previamente resueltos. Para el primer agente, su función de utilidad en cada uno de los estados del MDP asociado corresponde a la utilidad si sólo hubiese un recurso. Para el segundo agente, esta función se actualiza basándose en la política individual correspondiente al primer agente. La función de utilidad para este segundo agente se actualiza modificando las recompensas correspondientes a los estados visitados por el primer agente, reflejando la efectividad adicional que el sistema recibe si un segundo agente visita dicho estado, en comparación a sólo disponer de uno de ellos ahí.

En específico, para el caso de este trabajo, $\psi(k) = 1 - (1 - \xi)^k$. Así, $\mu(s_i)$ corresponde a la efectividad adicional si el recurso i se encuentra en el estado s_i . Esta efectividad adicional es

obtenida mediante el cálculo de la efectividad otorgada por el recurso i en el estado s_i , considerando la efectividad que aportan ahí los recursos previamente calculados, menos la efectividad de dichos recursos sin considerar a i en dicho estado. Por ejemplo, si dos recursos previamente calculados se encuentran en el estado s_i y un tercer recurso visita dicho estado, la utilidad individual para este tercer recurso no corresponde a la utilidad conjunta de tener tres recursos en dicho estado, si no que a la efectividad adicional de tener tres recursos en comparación a tener sólo dos.

4.1.3. Heurística JESP

Un algoritmo, muy cercano al enfoque greedy descrito, que garantiza obtener una política conjunta localmente óptima corresponde a: Búsqueda de Políticas Mediante Equilibrio Conjunto (JESP, por sus siglas en inglés) [48]. La idea central de este algoritmo es iterar sobre el conjunto de agentes obteniendo la política individual óptima para cada uno que maximiza la recompensa esperada conjunta, dejando fija las políticas individuales para el resto de los agentes. Este proceso se repite hasta que se alcanza un equilibrio en el cual las políticas individuales obtenidas no aumentan la recompensa esperada conjunta. Este equilibrio representa un óptimo local del problema y corresponde a un equilibrio de Nash [49]. Una característica de este enfoque es que su espacio de búsqueda de soluciones, correspondiente a políticas conjuntas, no se encuentra restringido.

Sea n el número de agentes. El siguiente pseudocódigo presenta el algoritmo descrito.

Algorithm 3 Heurística JESP

```

prev ← random joint policy
conv ← 0
while conv ≠  $n - 1$  do
  for  $i = 1 : n$  do
    fix policy for all agents except  $i$ 
    PolicySpace ← list of all policies for  $i$ 
    new ← BestPolicy( $i$ , PolicySpace, prev)
    if new.value = prev.value then
      conv ← conv + 1
    else
      prev ← new
      conv ← 0
    end if
  if conv =  $n - 1$  then
    break
  end if
  end for
end while
return new

```

Este algoritmo modifica la política correspondiente a un agente en cada iteración, manteniendo las políticas de los demás $n - 1$ agentes fijas. La función BestPolicy obtiene la política conjunta

que maximiza la utilidad esperada conjunta. Esto se realiza mediante la fijación de las políticas individuales correspondientes a $n - 1$ agentes, y buscando exhaustivamente en el espacio completo de políticas individuales el agente restante. Por lo tanto, en cada iteración del algoritmo, el valor de la política conjunta aumenta o permanece sin alteraciones. Este proceso es repetido hasta que se alcanza un equilibrio; en donde, las políticas individuales para los n agentes no se modifica. Este algoritmo garantiza la obtención de una política conjunta correspondiente a un máximo local, ya que el valor obtenido para dicha política en cada iteración no decrece.

4.1.4. Heurística Solución a-priori

Un enfoque para obtener una solución factible para el problema Dec-MDP consiste en particionar el grafo que representa las estaciones y sus respectivas conexiones entre ellas, en un grafo conexo de menor tamaño por cada agente defensivo disponible. A cada uno de dichos grafos se le asigna un agente, se mantiene las utilidades originales para cada estación y se considera el mismo número de períodos de tiempo que para el problema Dec-MDP original. Luego se asocia un problema MDP a cada uno de dichos grafos, y se resuelve mediante método estándar. Una vez que todos estos problemas han sido resueltos, la combinación de las políticas individuales obtenidas para cada agente corresponde a una solución factible para el problema Dec-MDP en cuestión. La Figura 4.1 muestra en color verde, a modo de ejemplo, la partición asociada a dos recursos defensivos que se encuentran en una red conformada por seis estaciones.

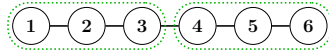


Figura 4.1: Ejemplo de partición del grafo para la utilización del enfoque basado en incorporación de solución a-priori.

4.1.5. Heurística MILP y Cota para Problema Dec-MDP

El modelo Dec-MDP utilizado en este trabajo admite la obtención de una cota superior para su valor objetivo mediante la formulación de un problema de programación lineal mixta. Para definir dicha formulación, sea L el conjunto de estaciones consideradas en el problema; y $\varrho(l)$ el conjunto de estaciones que están directamente conectadas a la estación $l \in L$. Además, se define $\delta(l) = \varrho(l) \cup l, \forall l \in L$; es decir, el conjunto que contiene a la estación $l \in L$ junto con las estaciones directamente conectadas a ella. Se consideran las siguientes variables de decisión. $x_{i,l,a}^t$ ($\forall i \in Ag, \forall l \in L, \forall a \in A_i, \forall t \in T$) corresponde a la probabilidad de que el agente i se encuentre en la estación l durante el período de tiempo t y ejecute ahí la acción a ; $w_{i,s,a}^t$ ($\forall i \in Ag, \forall s \in S_i, \forall a \in A_i, \forall t \in T$) a una variable binaria que indica si el agente i elige la acción a al estar en el estado s durante el período de tiempo t ; y $\gamma_{i,l}$ ($\forall i \in Ag, \forall l \in L$) a la probabilidad de que el agente i se encuentre en la estación l en el primer período de tiempo.

Para lograr la formulación del problema de programación lineal mixta mencionado, primero se

define el siguiente problema de programación no lineal que obtiene la política conjunta óptima para el problema Dec-MDP de interés.

$$\max_{(\mathbf{x}, \mathbf{w}, \gamma)} \mathbb{E}\{R(s)\} \quad (4.13)$$

s. a.

$$\sum_{a \in \delta(l)} x_{i,l,a}^1 = \gamma_{i,l}, \quad \forall i \in Ag, \forall l \in L \quad (4.14)$$

$$\sum_{a \in \delta(l)} x_{i,l,a}^t - (1 - p_d) \sum_{l' \in \delta(l)} x_{i,l',l}^{t-1} - p_d \sum_{a \in \delta(l)} x_{i,l,a}^{t-1} = 0, \quad \forall i \in Ag, \forall l \in L, \forall t \in \{2, \dots, T\} \quad (4.15)$$

$$x_{i,l,a}^t \leq w_{i,l,a}^t, \quad \forall i \in Ag, \forall l \in L, \forall a \in \delta(l), \forall t \in T \quad (4.16)$$

$$\sum_{l \in L} \gamma_{i,l} = 1, \quad \forall i \in Ag \quad (4.17)$$

$$\sum_{a \in \delta(l)} w_{i,l,a}^t = 1, \quad \forall i \in Ag, \forall l \in L, \forall t \in T \quad (4.18)$$

$$x_{i,l,a}^t \geq 0, \quad \forall i \in Ag, \forall l \in L, \forall a \in \delta(l), \forall t \in T \quad (4.19)$$

$$w_{i,l,a}^t \in \{0; 1\}, \quad \forall i \in Ag, \forall l \in L, \forall a \in \delta(l), \forall t \in T \quad (4.20)$$

$$\gamma_{i,l} \in \{0; 1\}, \quad \forall i \in Ag, \forall l \in L. \quad (4.21)$$

El problema de optimización presentado en las ecuaciones (4.13) a (4.21) corresponde a un problema cuya región factible está definida en términos de ecuaciones lineales; sin embargo, su función objetivo es no lineal. En específico, la ecuación (4.15) obliga a que la probabilidad de que un cierto agente pueda tomar alguna acción en una determinada estación sea igual a suma de la probabilidad de haber decidido estar en dicha acción en el período anterior y haberlo logrado más la probabilidad haber estado ahí, decidir cambiarse de estación y no haberlo logrado.

Con el propósito de obtener una aproximación lineal de dicho problema, la cual otorgue una cota

superior para su valor óptimo, es posible desarrollar su función objetivo de la siguiente manera:

$$\mathbb{E} \{R(s)\} = \mathbb{E} \left\{ \sum_{b \in B} y_b \cdot \text{eff}(s, b) \right\} \quad (4.22)$$

$$= \sum_{b \in B} y_b \cdot \mathbb{E} \{ \text{eff}(s, b) \} \quad (4.23)$$

$$= \sum_{b \in B} y_b \cdot \mathbb{E} \{ 1 - (1 - \xi)^{\sum_{i \in A_g} I_{s_i=b}} \} \quad (4.24)$$

$$\leq \sum_{b \in B} y_b \cdot \left(1 - (1 - \xi)^{\mathbb{E} \left\{ \sum_{i \in A_g} I_{s_i=b} \right\}} \right) \quad (4.25)$$

$$= \sum_{(l,t) \in B} y_b \cdot \left(1 - (1 - \xi)^{\sum_{i \in A_g} \sum_{a \in \delta(l)} x_{i,l,a}^t} \right) \quad (4.26)$$

$$= \sum_{(l,t) \in B} y_b \cdot \psi \left(\sum_{i \in A_g} \sum_{a \in \delta(l)} x_{i,l,a}^t \right). \quad (4.27)$$

En donde, $\psi(x) = 1 - (1 - \xi)^x$ es una función cóncava, como se mencionó en el Capítulo 3; y en la inecuación (4.25) se ha utilizado la desigualdad de Jensen, presentada en la ecuación (2.39), para el caso de funciones cóncavas.

Si bien la función (4.27) es cóncava en las variables de decisión para el problema definido en las ecuaciones (4.13) a (4.21), es posible obtener una aproximación superior lineal por tramos de ella. Sea $\bar{\psi}(x)$ la función que corresponde a dicha aproximación, la cual cumple:

$$\bar{\psi}(x) \approx 1 - (1 - \xi)^x \quad (4.28)$$

$$\bar{\psi}(x) \geq 1 - (1 - \xi)^x. \quad (4.29)$$

En donde (4.28) se refiere a que $\bar{\psi}(x)$ debe ser una función que se aproxime a $1 - (1 - \xi)^x$ para el dominio de interés.

Una función que cumple dichas hipótesis corresponde a:

$$\bar{\psi}(x) = \min_{i \in \{1, \dots, m\}} \psi'(p_i)x + b_i; \quad (4.30)$$

en donde,

$$\psi'(p_i) = -(1 - \xi)^{p_i} \cdot \ln(1 - \xi), \quad (4.31)$$

$$b_i = 1 - (1 - \xi)^{p_i} - \psi'(p_i) \cdot p_i, \quad (4.32)$$

y $\{p_1, \dots, p_m\}$ corresponde a un conjunto de puntos en el dominio de interés de la función $\psi(\cdot)$.

Con esto, es posible obtener una cota superior del valor óptimo para el problema Dec-MDP de interés mediante la formulación del siguiente problema de programación lineal mixto:

$$\max_{(\mathbf{x}, \mathbf{w}, \gamma, \mathbf{z})} \sum_{(l,t) \in B} y_{l,t} \cdot z_{l,t} \quad (4.33)$$

s.a.

$$\sum_{a \in \delta(l)} x_{i,l,a}^1 = \gamma_{i,l}, \quad \forall i \in Ag, \forall l \in L \quad (4.34)$$

$$\sum_{a \in \delta(l)} x_{i,l,a}^t - (1 - p_d) \sum_{l' \in \delta(l)} x_{i,l',l}^{t-1} - p_d \sum_{a \in \delta(l)} x_{i,l,a}^{t-1} = 0, \quad \forall i \in Ag, \forall l \in L, \forall t \in \{2, \dots, T\} \quad (4.35)$$

$$x_{i,l,a}^t \leq w_{i,l,a}^t, \quad \forall i \in Ag, \forall l \in L, \forall a \in \delta(l), \forall t \in T \quad (4.36)$$

$$\sum_{l \in L} \gamma_{i,l} = 1, \quad \forall i \in Ag \quad (4.37)$$

$$\sum_{a \in \delta(l)} w_{i,l,a}^t = 1, \quad \forall i \in Ag, \forall l \in L, \forall t \in T \quad (4.38)$$

$$z_{l,t} \leq \psi'(p_j) \left(\sum_{i \in Ag} \sum_{a \in \delta(l)} x_{i,l,a}^t \right) + b_j, \quad \forall l \in L, \forall t \in T, \forall j \in \{1, \dots, m\} \quad (4.39)$$

$$x_{i,l,a}^t \geq 0, \quad \forall i \in Ag, \forall l \in L, \forall a \in \delta(l), \forall t \in T \quad (4.40)$$

$$w_{i,l,a}^t \in \{0; 1\}, \quad \forall i \in Ag, \forall l \in L, \forall a \in \delta(l), \forall t \in T \quad (4.41)$$

$$\gamma_{i,l} \in \{0; 1\}, \quad \forall i \in Ag, \forall l \in L. \quad (4.42)$$

$$z_{l,t} \geq 0, \quad \forall l \in L, \forall t \in T. \quad (4.43)$$

Proposición 2: una cota superior para el valor objetivo del problema definido en (4.13) a (4.21) corresponde al valor objetivo del problema definido en (4.33) a (4.43).

DEMOSTRACIÓN. Sea $(\mathbf{x}_1^*, \mathbf{w}_1^*, \gamma_1^*)$ y $(\mathbf{x}_2^*, \mathbf{w}_2^*, \gamma_2^*, \mathbf{z}_2^*)$ soluciones de los problema definido por las ecuaciones (4.13) a (4.21) y (4.33) a (4.43), respectivamente.

$$\text{Sea } (\mathbf{x}_1^*, \mathbf{w}_1^*, \gamma_1^*, \mathbf{z}); \text{ con } z_{l,t} = \min_{j \in \{1, \dots, m\}} \psi'(p_j) \left(\sum_{i \in Ag} \sum_{a \in \delta(l)} x_{i,l,a}^{1*} \right) + b_j, \quad \forall l \in L, \forall t \in T.$$

Se tiene que $(\mathbf{x}_1^*, \mathbf{w}_1^*, \gamma_1^*, \mathbf{z})$ es solución factible para el problema definido por las ecuaciones (4.33) a (4.43).

Los valores objetivos de los problemas (4.13) a (4.21) y (4.33) a (4.43) corresponden a $\mathbb{E} \{R(s_{\mathbf{x}_1^*, \mathbf{w}_1^*, \gamma_1^*})\}$ y $\sum_{(l,t) \in B} y_{l,t} \cdot z_{2,l,t}^*$, respectivamente.

Se tiene que,

$$\begin{aligned}
\mathbb{E} \{R(s_{\mathbf{x}_1^*, \mathbf{w}_1^*, \gamma_1^*})\} &\leq \sum_{(l,t) \in B} y_b \cdot \psi \left(\sum_{i \in A_g} \sum_{a \in \delta(l)} x_{1i,l,a}^{*t} \right) \\
&\leq \sum_{(l,t) \in B} y_b \cdot \left\{ \min_{j \in \{1, \dots, m\}} \psi'(p_j) \left(\sum_{i \in A_g} \sum_{a \in \delta(l)} x_{1i,l,a}^{*t} \right) + b_j \right\} \\
&= \sum_{(l,t) \in B} y_b \cdot z_{l,t} \\
&\leq \sum_{(l,t) \in B} y_b \cdot z_{2l,t}^*
\end{aligned}$$

En donde $s_{\mathbf{x}, \mathbf{w}, \gamma}$ corresponde al estado global del sistema inducido por la solución $\mathbf{x}, \mathbf{w}, \gamma$. \square

Además, es posible obtener una heurística para el problema Dec-MDP descrito en las ecuaciones (4.13) a (4.21). Esto se realiza mediante la resolución del problema de programación lineal mixta presentado en las ecuaciones (4.33) a (4.43), y evaluando las correspondientes variables de decisión óptimas obtenidas en la ecuación (4.22). Con esto, se consigue una solución factible para el problema Dec-MDP, la cual se evalúa en su función objetivo correspondiente.

También, es posible obtener una cota superior y una heurística para este dicho problema Dec-MDP mediante la resolución de la relajación lineal del problema descrito. Si bien esta cota superior es de peor calidad que la obtenida mediante la formulación MILP, la resolución del problema asociado para obtenerla es más sencilla.

Sea η^* el valor óptimo de un problema Dec-MDP asociado a la resolución del subproblema del juego de seguridad de interés en este trabajo. Sea η_{mip}^* y η_{lp}^* la cota superior obtenida mediante la formulación MILP presentada y su relajación lineal, respectivamente. Es posible modificar la cota superior presentada en la ecuación (3.32) de la siguiente manera:

$$v^* \leq v_{RM} + \eta^* - z \tag{4.44}$$

$$\leq v_{RM} + \eta_{mip}^* - z \tag{4.45}$$

$$\leq v_{RM} + \eta_{lp}^* - z. \tag{4.46}$$

4.2. Comparación de Heurísticas para Resolver Dec-MDP

Con el objetivo de poder comparar las heurísticas para resolver Dec-MDP presentadas, y enfocándose en la resolución del subproblema del modelo de generación de columnas detallado en el Capítulo 3, se han considerado instancias de pruebas correspondientes a diversas configuraciones para una red de metro. Para esto, se ha diseñado seis conjuntos de instancias. Cada uno de estos

conjuntos está conformado por instancias que poseen en común el mismo número de estaciones, períodos de tiempo considerados, y recursos defensores. Estos números definen los parámetros que caracterizan las instancias de un determinado conjunto. Sin embargo, la estructura del grafo que define las conexiones entre las estaciones que conforman la red de metro es diferente para cada una de las instancias pertenecientes a un mismo conjunto. Así mismo, una determinada instancia perteneciente a un conjunto posee utilidades, en cada una de sus estaciones y períodos de tiempo, diferentes a los de otra instancia perteneciente al mismo conjunto.

Se refiere a estos conjuntos con una numeración desde el número 1 al 6, en donde el conjunto número 1 corresponde al grupo de instancias cuyos valores para los parámetros que las caracterizan son los menores. El valor de estos parámetros va aumentando a medida que aumenta la numeración para el conjunto. La Tabla 4.1 describe el valor de los parámetros que caracteriza a las instancias que conforman cada uno de los conjuntos mencionados. Además, se refiere a los estados que posee una determinada instancia como las distintas tuplas conformadas por las estaciones y períodos de tiempo de dicha instancia. Así, la cantidad de estados para las instancias pertenecientes a un mismo conjunto es constante.

	Número de Estaciones	Períodos de Tiempo	Recursos Defensores
Conjunto Nº1	6	5	2
Conjunto Nº2	8	8	3
Conjunto Nº3	10	8	4
Conjunto Nº4	12	10	5
Conjunto Nº5	14	10	5
Conjunto Nº6	16	12	6

Tabla 4.1: Parámetros que caracterizan a las instancias de cada conjunto.

La estructura de los grafos para las instancias de un mismo conjunto consiste en un nodo por cada estación que caracterice a la instancia. Sea n el número de estaciones considerado; entonces, se dispone de n nodos, numerados desde 1 a n . Los arcos entre estaciones vienen dados por una estructura común para todas las instancias de un determinado conjunto, y una estructura estocástica propia de cada instancia en particular. La estructura común corresponde a considerar los primeros $\lfloor (n - 1)/2 \rfloor$ nodos, y establecer un arco entre cada uno de estos nodos ($\forall i \in 1, \dots, \lfloor (n - 1)/2 \rfloor$) y el nodo correspondiente al doble de su numeración ($2 * i$); similarmente, se añade un arco entre cada uno de los nodos previamente considerados y el nodo correspondiente al sucesor del doble de su numeración ($2 * i + 1$). En caso que el número de nodos totales sea par, se añade un arco entre el nodo que corresponde a la mitad de la numeración y el último nodo. De esta forma es posible obtener un grafo conexo para cualquier número de estaciones considerado.

En caso que la instancia este conformada por un valor igual o superior a 20 estaciones, la estructura estocástica consiste en elegir de forma uniformemente distribuida 10 pares distintos de nodos que no se encuentren conectados mediante la estructura común, y establecer un arco entre ellos. Se descarta unir una estación con sigio misma. En caso de que la instancia posea un número menor a 20 estaciones, se realiza el mismo procedimiento descrito, pero seleccionando un número de $\lfloor n/2 \rfloor$ pares de nodos no conectados mediante la estructura común. La utilización de una estructura

estocástica logra la generación de distintos grafos conexos, permitiendo tener variabilidad en los grafos de las instancias que pertenecen a un mismo conjunto. Por otro lado, la utilización de una estructura común permite seccionar los grafos de instancias pertenecientes a un mismo conjunto en grafos más simples y comunes para dichas instancias. Esto será de utilidad para la aplicación de la heurística de enfoque greedy con solución a-priori a instancias de un mismo conjunto.

Las Figuras 4.2 a 4.7 presentan, a modo de ejemplo, un posible grafo para una instancia perteneciente a cada uno de los conjuntos considerados. La estructura común se presenta mediante las líneas continuas en color negro, mientras que la estructura estocástica se presenta utilizando líneas discontinuas en color azul.

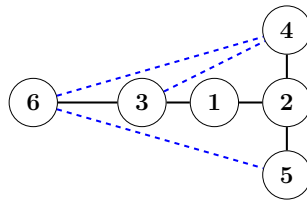


Figura 4.2: Ejemplo de grafo para una instancia perteneciente al conjunto N°1.

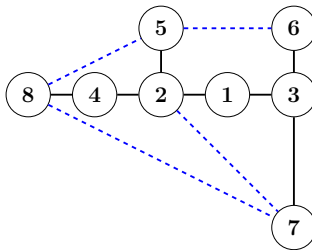


Figura 4.3: Ejemplo de grafo para una instancia perteneciente al conjunto N°2.

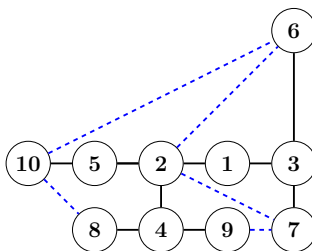


Figura 4.4: Ejemplo de grafo para una instancia perteneciente al conjunto N°3.

La generación de la utilidad para cada una de las estaciones y períodos de tiempo de una instancia se realiza de forma aleatoria. Para esto, se considera cuatro distribuciones distintas. Para cada una de las instancias se elige una de estas distribuciones en particular y se extraen de ella en forma independiente la utilidad para cada una de las estaciones y períodos de tiempo que conforman la

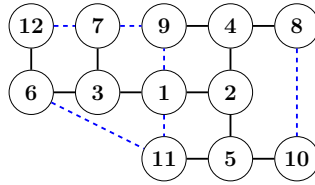


Figura 4.5: Ejemplo de grafo para una instancia perteneciente al conjunto N°4.

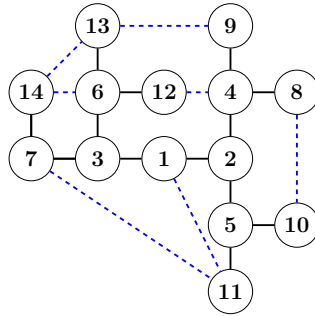


Figura 4.6: Ejemplo de grafo para una instancia perteneciente al conjunto N°5.

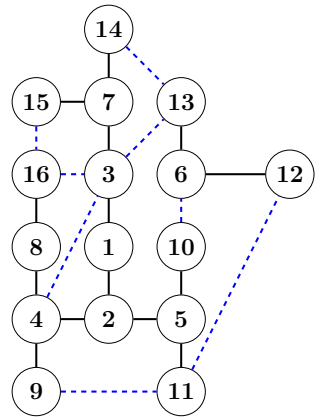


Figura 4.7: Ejemplo de grafo para una instancia perteneciente al conjunto N°6.

instancia. Tres de estas distribuciones corresponden a una distribución uniforme. La primera de ellas toma valores en el intervalo $[0; 10]$, la segunda en el intervalo $[3; 7]$, y la tercera en $[4.5; 5.5]$. De esta forma, se tiene tres distribuciones uniformes cuyos valores medios corresponden a un valor común de 5. Su diferencia radica en la desviación estándar que presentan. Siendo esta desviación mayor para la primera distribución, intermedia para la segunda, y menor para la tercera. La cuarta distribución corresponde a elegir un cierto número de estados, el cual es común para todas las instancias de un mismo conjunto y se representa por ψ , que conforman una cierta instancia y generar para cada uno de ellos, de forma independiente, su utilidad mediante una distribución uniforme en el intervalo $[0; 10]$. Para el resto de los estados, su utilidad se genera, de forma independiente, con un distribución uniforme en el intervalo $[0; 0.1]$.

Esta cuarta distribución para las utilidades de los estados de una instancia se inspira en las utilidades que puede tener el subproblema que se quiere resolver mediante las heurísticas presentadas en este capítulo. En específico, se quiere resolver un Dec-MDP cuya función a maximizar está dada por la ecuación (3.16). En esta ecuación, la utilidad para cada uno de los estados viene dada por las variables duales del problema maestro asociado al problema presentado en las ecuaciones (3.8) a (3.13). En donde estas variables pueden ser nulas para algunos casos, y en otros tener un valor positivo.

EL valor de ψ para un conjunto de instancias en particular corresponde al 20 % de la cantidad de estados que poseen las instancias pertenecientes a dicho conjunto. En el caso que dicho porcentaje no corresponda a un número entero, se considera el cajón superior de su valor. La Tabla 4.2 detalla el valor de ψ para cada uno de los conjuntos considerados.

	Conjunto Nº1	Conjunto Nº2	Conjunto Nº3	Conjunto Nº4	Conjunto Nº5	Conjunto Nº6
ψ	6	13	16	24	28	39

Tabla 4.2: Valor de ψ para cada conjunto de instancias considerado.

Para cada una de las instancias consideradas, y para el resto de este trabajo, se considera un valor de la efectividad de cada recurso, ξ , igual a 0.4; y que la probabilidad, para cada uno de los recursos, de fracasar al realizar una determinada acción igual a 0.1.

Para el desarrollo de los experimentos computacionales de este trabajo, presentados en los Capítulos 4 y 5, se utilizó Java. Los modelos de optimización fueron resueltos con la utilización de IBM ILOG Cplex.

4.2.1. Elección de Parámetros para Heurística Entropía Cruzada

La heurística basada en el método de entropía cruzada presentada en este capítulo requiere de cuatro parámetros para su utilización: α , I , N y ρ . Con el objetivo de ajustar estos parámetros para utilizar dicha heurística en la problemática de interés de este trabajo, se diseña una metodología que hace uso de los conjuntos de instancias presentados. Esta metodología, para cada uno de los conjuntos de instancias considerados, ajusta el valor de los parámetros mediante un proceso iterativo, comenzando con una asignación inicial para el valor de cada uno de ellos.

Para un determinado conjunto de instancias, se considera ciertos valores para el parámetro α , y para cada uno de estos valores se genera una determinada cantidad de instancias por cada una de las cuatro distribuciones para las utilidades descritas. Cada una de estas instancias, correspondiente a un problema Dec-MDP, es resuelta mediante la heurística en cuestión utilizando el correspondiente parámetro α y el valor inicial para los demás parámetros. Los resultados obtenidos para cada parámetro α se comparan entre sí mediante intervalos de confianza, y se elige el que

presente el mayor valor promedio estadísticamente significativo. El valor elegido para este parámetro actualiza su valor inicial. Este procedimiento es realizado a continuación para el parámetro I , considerando la actualización de los parámetros previamente calculados. Luego se realiza dicho procedimiento para el parámetro N , y finalmente para el parámetro ρ . Los intervalos de confianza usados aquí, y en el resto de este trabajo, son calculados utilizando un nivel de confianza de 90 %.

El detalle de la aplicación de la metodología descrita a los conjuntos de instancias considerados se presenta en la Sección Anexos de este trabajo. La Tabla 4.3 resume los valores finales obtenidos de los parámetros requeridos por la heurística para instancias correspondientes al conjunto Nº1.

	α	I	N	ρ
Conjunto Nº1	0.2	90	50	0.3

Tabla 4.3: Valor de los parámetros requeridos por la heurística Entropía Cruzada para instancias correspondientes al conjunto Nº1.

Para los conjuntos de instancias considerados, esta metodología sólo se aplicó al primero de ellos. Para conjuntos cuyas instancias son de mayor tamaño, el tiempo requerido para obtener el resultado de la metodología es demasiado alto.

4.2.2. Comparación de Heurísticas

Una vez que se ha presentado diversas heurísticas para la resolución de un problema Dec-MDP, se está en condiciones de evaluar cuál de ellas es la que mejor desempeño otorga para ser utilizada dentro del enfoque de generación de columnas presentado para la resolución del juego de seguridad de interés. Para esto, las heurísticas que se consideran corresponden a la heurística Entropía Cruzada, heurística MILP, y una obtenida mediante la combinación de la heurística Greedy, la heurística JESP y la heurística Solución a-priori (descritas en las subsecciones 4.1.1, 4.1.5, 4.1.2, 4.1.3 y 4.1.4; respectivamente). Esta combinación consiste en, para una cierta instancia, primero utilizar la heurística Solución a-priori mediante el particionamiento del grafo que describe dicha instancia. Y luego, utilizar la heurística JESP, usando el resultado de Solución a-priori como su solución inicial. Además, se considera una variante, en donde la solución inicial corresponde a la solución otorgada por la heurística Greedy, en vez de utilizar la otorgada por la heurística Solución a-priori.

La forma en que dichas heurísticas son utilizadas se describe a continuación, y se referirá a cada una de dichas formas de utilización como un método heurístico de aquí en adelante. Para el caso de la heurística Entropía Cruzada, sus parámetros corresponden a los presentados en la Tabla 4.3, los cuales fueron obtenidos utilizando la metodología descrita. Cada una de las instancias consideradas es resuelta diez veces mediante dicha heurística, logrando la obtención de cuatro métodos heurísticos. El primero, nombrado D1 corresponde al resultado de la primera resolución; el segundo, D3, corresponde al mejor resultado de las tres primeras soluciones. Análogamente, se considera los métodos D5 y D10. El hecho de considerar dichos métodos para esta heurística se

debe a que, como su resultado es estocástico, utilizarla para resolver múltiples veces una determinada instancia puede resultar en la obtención de diferentes resultados. El método heurístico que corresponde a la resolución mediante la heurística MILP se denota por LbMip. Por otro lado, el método correspondiente a la resolución mediante la heurística JESP, comenzando con la solución otorgada por la heurística Solución a-priori, se denota por Warm Start; mientras que si comienza con la otorgada por la heurística Greedy, se denota por Cold Start.

Para comparar estos métodos heurísticos, de forma similar a lo hecho en la metodología para ajustar los parámetros de la heurística Entropía Cruzada, se hace uso de los seis conjuntos de instancias descritos. Para cada uno de estos conjuntos, se generan 5000 instancias aleatorias por cada una de las cuatro distribuciones para las utilidades consideradas. Así, para cada uno de dichos conjuntos se generan 20000 instancias aleatorias, en donde cada una de ellas corresponde a un problema Dec-MDP. Cada una de estas instancias es resuelta mediante el uso de los métodos heurísticos mencionados. Además, la comparación de los resultados obtenidos se realiza de igual forma a lo hecho en la metodología de ajuste de parámetros mencionada. Es decir, para cada conjunto de instancias considerado, los valores promedio de los resultados obtenidos al resolver las instancias con cada uno de los métodos heurísticos presentados son comparados entre sí mediante intervalos de confianza.

Las Figuras 4.8 a 4.13 presentan en color verde la partición de la estructura común de los grafos correspondientes a cada uno de los conjuntos de instancias considerados para la utilización del enfoque que utiliza incorporación de solución a-priori.

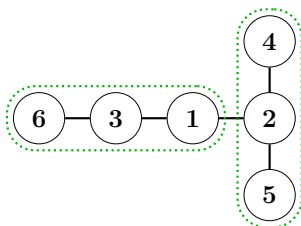


Figura 4.8: Partición de la estructura común del grafo correspondiente a una instancia del conjunto N°1.

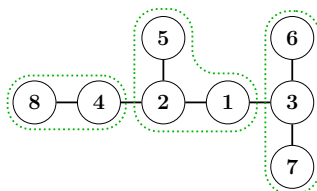


Figura 4.9: Partición de la estructura común del grafo correspondiente a una instancia del conjunto N°2.

La Tabla 4.4 presenta los resultados obtenidos al aplicar los métodos heurísticos descritos a las instancias aleatorias correspondientes al conjunto N°1. La primera fila de dicha tabla contiene

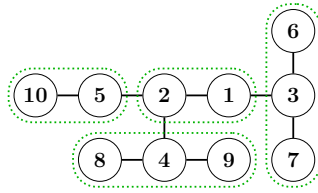


Figura 4.10: Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº3.

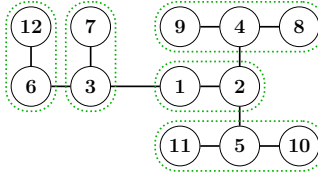


Figura 4.11: Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº4.

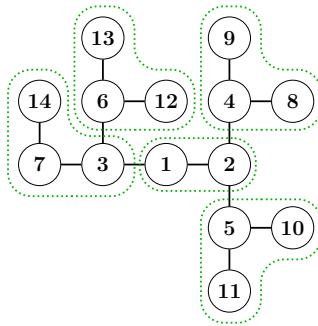


Figura 4.12: Partición de la estructura común del grafo correspondiente a una instancia del conjunto Nº5.

el valor promedio de los resultados obtenidos con cada método. Las demás celdas presentan el intervalo de confianza que se obtiene al hacer la resta de los valores obtenidos por el método correspondiente a la columna de dicha celda menos los obtenidos por el método correspondiente a su fila.

La Tabla 4.5 presenta estadísticas del tiempo requerido para la resolución de las instancias pertenecientes al conjunto Nº1 mediante el uso de los métodos heurísticos descritos. Además, dicha tabla presenta las estadísticas del tiempo requerido para el cálculo de la cota superior del valor óptimo de las instancias pertenecientes a dicho conjunto mediante el uso de la formulación MILP descrita, el cual es denotado por $UbMip$. Junto con esto, se presenta estadísticas del tiempo de cálculo de la relajación lineal de dicho MILP, cuyo valor óptimo también corresponde a una cota superior para el respectivo problema Dec-MDP. Este último tiempo se denota por $UbLp$.

Se aprecia que al comparar los valores obtenidos al resolver las instancias pertenecientes al con-

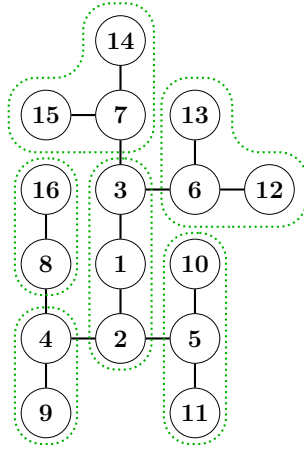


Figura 4.13: Partición de la estructura común del grafo correspondiente a una instancia del conjunto N°6.

	D1	D3	D5	D10	Cold Start	Warm Start	LbMip
Promedio	24.4580	24.5296	24.5454	24.5579	24.4895	24.4990	24.3963
D1	[0; 0]	[0.0696; 0.0735]	[0.0853; 0.0894]	[0.0977; 0.1019]	[0.0286; 0.0343]	[0.0380; 0.0438]	[-0.0643; -0.0593]
D3	[-0.0735; -0.0696]	[0; 0]	[0.0152; 0.0163]	[0.0275; 0.0290]	[-0.0424; -0.0379]	[-0.0330; -0.0283]	[-0.1351; -0.1316]
D5	[-0.0894; -0.0853]	[-0.0163; -0.0152]	[0; 0]	[0.0120; 0.0129]	[-0.0582; -0.0537]	[-0.0487; -0.0441]	[-0.1508; -0.1475]
D10	[-0.1019; -0.0977]	[-0.0290; -0.0275]	[-0.0129; -0.0120]	[0; 0]	[-0.0706; -0.0661]	[-0.0612; -0.0566]	[-0.1632; -0.1599]
Cold Start	[-0.0343; -0.0286]	[0.0379; 0.0424]	[0.0537; 0.0582]	[0.0661; 0.0706]	[0; 0]	[0.0069; 0.0121]	[-0.0957; -0.0907]
Warm Start	[-0.0438; -0.0380]	[0.0283; 0.0330]	[0.0441; 0.0487]	[0.0566; 0.0612]	[-0.0121; -0.0069]	[0; 0]	[-0.1053; -0.1001]
LbMip	[0.0593; 0.0643]	[0.1316; 0.1351]	[0.1475; 0.1508]	[0.1599; 0.1632]	[0.0907; 0.0957]	[0.1001; 0.1053]	[0; 0]

Tabla 4.4: Valor promedio e intervalo de confianza (método columna menos método fila) para instancias correspondientes al conjunto N°1.

junto N°1, el método que presenta mejor desempeño corresponde a D10; ya que su valor promedio es superior a dicho valor para los demás métodos, y todos los intervalos de confianza obtenidos para D10 corresponden a un intervalo cuyos valores extremos son positivos. Sin embargo, este método presenta el peor desempeño en cuanto al tiempo requerido para proporcionar una solución.

Al observar los tiempos requeridos para los métodos que utilizan la heurística Entropía Cruzada (D1, D3, D5 y D10) y el que utiliza la heurística MILP (LbMip), se aprecia que estos son órdenes de magnitud mayores que el requerido por los métodos que utilizan la heurística JESP. Esto hace que la elección del método heurístico para el conjunto N°1 quede restringido a Warm

	Promedio [ms]	Desviación Estándar [ms]
D1	1420	308
D3	4198	515
D5	6962	691
D10	13868	1164
Cold Start	0.3376	1.3239
Warm Start	0.3669	1.3892
LbMip	77923	133440
UbMip	77923	133440
UbLp	6.0613	8.554

Tabla 4.5: Tiempos de resolución de los diversos métodos utilizados para instancias correspondientes al conjunto N°1.

Start o Cold Start. Ya que sus tiempos de resolución son comparables, y que Warm Star presenta mejor desempeño en cuanto los resultados obtenidos, este método es elegido para dicho conjunto.

Dado que los tiempos requeridos por los métodos D1, D3, D5, D10 y LbMip son demasiado grandes, en comparación a los requeridos por Warm Start y Cold Start, estos primeros métodos mencionados son descartados para su uso en conjuntos instancias de mayor tamaño. Este descarte se justifica ya que lo que se busca es una heurística para ser aplicada en un subproblema de generación de columnas, en donde el tiempo requerido para su resolución es crítico. Así mismo, se descarta el cálculo de la cota superior mediante la formulación MILP, dejando sólo el cálculo mediante su relajación lineal.

La Tabla 4.6 presenta los resultados obtenidos al utilizar los métodos heurísticos considerados en los conjuntos de instancias N°2, N°3, N°4, N°5 y N°6. C.S.V.P. y W.S.V.P corresponden al valor promedio de los resultados obtenidos mediante el método Cold Start y Warm Start, respectivamente. El intervalo de confianza se obtiene mediante la resta de los valores obtenidos con el método Cold Start menos los obtenidos con Warm Start. Además, esta tabla presenta las estadísticas para el tiempo requerido por dichos métodos, y para la obtención de la cota superior mediante la formulación lineal, en los conjuntos de instancias mencionados. C.S.T.P. y C.S.D.E. corresponden al promedio y desviación estándar del tiempo requerido por el método Cold Start; W.S.T.P. y W.S.D.E. corresponden el promedio y desviación estándar del tiempo requerido por Warm Start; y UbLp T.P. y UbLp D.S. al promedio y desviación estándar del tiempo requerido para la obtención de la cota superior.

Se aprecia que para cada uno de los conjuntos N°2, N°3, N°4, N°5 y N°6, el método que presenta mejor resultado corresponde a Warm Start. Por otro lado, el tiempo de resolución promedio de este método para algunos conjuntos es mayor y para otros menor en comparación al de Cold Start. Sin embargo, los tiempos requeridos por dichos métodos son comparables, ya que sus estadísticas son del mismo orden de magnitud. Así, el método Warm Start es elegido para dichos conjuntos.

Con esto, el método que se elige para la resolución de instancias pertenecientes a los conjuntos

	C.S. V.P.	W.S. V.P.	Int. de Conf.	C.S. T.P. [ms]	C.S. D.E. [ms]	W.S. T.P. [ms]	W.S. D.E. [ms]	UbLp T.P. [ms]	UbLp D.E. [ms]
Cjto. Nº2	57.7075	57.7644	[-0.0626; -0.0511]	2.1162	6.5585	2.0845	6.6908	58.033	30.409
Cjto. Nº3	76.2455	76.3155	[-0.0772; -0.0629]	4.3178	9.1357	3.7902	7.5661	178.52	89.678
Cjto. Nº4	118.1723	118.3464	[-0.1841; -0.1642]	14.393	104.64	12.326	214.07	930.51	469.4
Cjto. Nº5	121.1741	121.3145	[-0.1509; -0.1299]	18.255	323.26	12.332	148.8	1272.9	681.08
Cjto. Nº6	173.4043	173.6646	[-0.2738; -0.2468]	39.108	101.85	29.669	145.81	27869	17249

Tabla 4.6: Estadísticas del valor y tiempo de los métodos utilizados para los conjuntos de instancias Nº2, Nº3, Nº4, Nº5 y Nº6.

Nº1, Nº2, Nº3, Nº4, Nº5 y Nº6 corresponde al método Warm Start. Se aprecia que, para cada uno de estos conjuntos de instancias, el tiempo requerido por dicho método heurístico es órdenes de magnitud menor que el tiempo empleado para la obtención su respectiva cota superior.

Capítulo 5

Desarrollo de Experimentos Computacionales para Resolución del Modelo

Una vez que se ha hecho el estudio de las heurísticas presentadas para resolución de Dec-MDP en el capítulo anterior, y se ha determinado cual es la que posee mejor desempeño en cuanto a su valor objetivo y tiempo de resolución para los conjuntos de instancias considerados, ésta heurística es elegida y se hace uso de ella para adaptar el enfoque de resolución del modelo de juego de seguridad presentado en el Capítulo 3. Además, en el presente capítulo se describen enfoques de resolución alternativos para dicho juego de seguridad, los cuales se basan en el enfoque mencionado. Con el objetivo de comparar el desempeño de estos enfoques, se presentan experimentos que permiten evaluarlos. Finalmente, se presenta un caso de estudio basado en una red de metro real, junto con su resolución mediante el uso de los enfoques desarrollados.

5.1. Metodologías de Resolución para el Modelo de Juego de Seguridad

El uso de la heurística elegida permite la adaptación del enfoque de resolución para el juego de seguridad mencionado. Dicha adaptación utiliza el algoritmo de múltiples problemas lineales para iterar sobre todas las posibles estrategias puras para el atacante. Para cada una de dichas iteraciones se resuelve un LP mediante la utilización de generación de columnas; en donde, el subproblema corresponde a un Dec-MDP, cuya resolución se realiza ahora de forma subóptima mediante el uso de la mencionada heurística elegida. Una vez que se obtienen dichas soluciones subóptimas en cada una de las iteraciones del algoritmo de múltiples problemas lineales, se elige la de mayor valor, la que representa una aproximación para el óptimo del modelo de juego de seguridad mencionado. A pesar de que dicha solución corresponde a un valor subóptimo, es posible obtener una garantía para ella. Esto se logra mediante el uso de la cota superior para el problema maestro presentada en la ecuación (3.32). La forma en que dicha cota es utilizada, para el presente trabajo, consiste en calcularla al término de la generación de columnas en cada una de las iteraciones del algoritmo de

múltiples problemas lineales, y elegir la mayor de todas ellas. Este valor corresponde a la garantía mencionada, ya que el valor óptimo para el modelo en cuestión a lo sumo corresponde al valor de dicha garantía.

Con el propósito de evaluar el desempeño de este enfoque de resolución para el modelo de juego de seguridad de interés, se hace uso de los seis conjuntos de instancias descritos en el capítulo anterior. Para cada uno de estos conjuntos se generan instancias, con sus correspondientes parámetros, y se utiliza la estructura respectiva para sus correspondientes grafos. Sin embargo, dado que ahora se pretende resolver dicho juego de seguridad, se requiere de pagos asociados a cada estación y para cada período de tiempo considerado, tanto para el defensor y para el atacante.

Estos pagos son generados de la siguiente manera. En cada una de las instancias mencionadas, para cada estación en el primer período de tiempo, se generan aleatoriamente los pagos tanto para el defensor como para el atacante. La distribución que se utiliza corresponde a valores enteros uniformemente distribuidos dentro de un intervalo, el cual varía dependiendo de si el jugador es el defensor o el atacante y si el pago corresponde a su beneficio o penalidad. Para el siguiente período de tiempo, en cada estación se generan los pagos de forma similar. La única salvedad es que el intervalo considerado para la distribución está definido, para cada tipo de pago y jugador, entre el entero antecesor y sucesor del valor generado, para el correspondiente pago y jugador, en dicha estación para el período de tiempo anterior. En el caso de que se genere un valor fuera del rango de su respectiva distribución para el primer período, se satura dicho valor al extremo más cercano del intervalo correspondiente. Este proceso se repite para todas las estaciones hasta el último período de tiempo considerado. Este mecanismo permite que la generación de los pagos, para una determinada estación, evolucione de forma llana a lo largo del horizonte temporal. De esta forma, se rescata la idea de que los pagos para una estación están influenciados por el afluente de gente que se encuentra ahí, el cual no varía de forma abrupta dentro de un intervalo temporal. Para este trabajo, los intervalos utilizados en el primer período de tiempo corresponden a $[0; 10]$ para el defensor y $[-10; 0]$ para el atacante.

Una particularidad interesante del enfoque de resolución para el modelo presentado consiste en que permite ser modificado con el propósito de no requerir de la generación de todas sus columnas originales. Esta modificación consiste en detener la generación de columnas para una determinada iteración del algoritmo de múltiples problemas lineales cuando se logre garantizar que generar columnas adicionales para dicha iteración no logrará obtener la solución factible con mayor valor objetivo para el problema total. Esta garantía se logra mediante el uso de una cota superior para el valor óptimo del LP correspondiente al problema maestro que se resuelve en cada iteración del algoritmo mencionado. Este nuevo enfoque consiste en almacenar el mayor valor de las soluciones factibles para los problemas maestros resueltos en las iteraciones previas del algoritmo, y descartar el problema maestro actual en caso de que alguna de sus correspondientes cotas superiores, obtenidas a lo largo de su respectivo proceso de generación de columnas, indique que su valor óptimo no supera al valor de dicha solución almacenada. En caso contrario, el proceso de generación de columnas no es interrumpido, obteniendo la mayor solución factible para el actual problema maestro; y en caso de ser mayor al valor de la actual solución almacenada, dicha solución almacenada es actualizada con el valor recientemente obtenido.

Este nuevo enfoque, si bien garantiza la obtención del mismo valor objetivo que se consigue al emplear el método de resolución presentado al comienzo de este capítulo, su ventaja sólo se traduce en un menor número de columnas generadas en total. El hecho de generar un menor número de columnas implica una menor cantidad de problemas Dec-MDP por resolver; y por lo tanto, menor tiempo dedicado a esta tarea particular dentro del enfoque de resolución del modelo. Sin embargo, la generación de cada una de dichas cota superiores también requiere tiempo, el cual es órdenes de magnitud mayor en comparación al que se requiere para la generación de una columna en el respectivo problema, tal como se mencionó en el capítulo anterior. Esto hace que dicho enfoque, si bien es conceptualmente útil, no sea atractivo de ser desarrollado. Sin embargo, una reducción en el tiempo requerido para la obtención de la cota superior mencionada o la cantidad de veces que se requiere su cálculo, podría hacer que este enfoque sea de utilidad para el caso de estudio del presente trabajo. Tomando esto en cuenta, se ha diseñado dos enfoques alternativos al enfoque que utiliza el algoritmo de múltiples problemas lineales descrito para la resolución del modelo de juego de seguridad de interés. Estos enfoques alternativos se basan en la idea presentada de detener el proceso de generación de columnas, desarrollado en cada una de las iteraciones del algoritmo mencionado, en algún momento en el que se garantice que la generación de columnas adicionales no es de utilidad.

5.1.1. Enfoque Basado en Número Restringido de Cotas Superiores

El primero de dichos enfoques consiste en sólo calcular la cota superior un número limitado de veces, para cada uno de los problemas maestros correspondientes a las estrategias puras del atacante. Además, los momentos, dentro del esquema de generación de columnas, en los que dicha cota superior es calculada son elegidos estratégicamente, para cada uno de los problemas maestros mencionados. Para esto, el enfoque presentado requiere de dos parámetros. El primero, α , indica el máximo número de veces que se puede calcular la cota superior dentro de la resolución de un problema maestro. Y el segundo, β , que corresponde al mínimo número de columnas que se deben haber generado, para cada problema maestro, luego de la obtención de una cota superior para volver a recalcular dicha cota. Este último parámetro asegura de que en cada problema maestro resuelto exista un intervalo entre las columnas generadas en las que se realiza el cálculo de la cota superior.

Para desarrollar esto, se considera el promedio del número de columnas que se generan en cada uno de los problema maestro correspondientes a las instancias pertenecientes a un mismo conjunto. Para determinar dicho promedio, para cada uno de los conjuntos considerados, se generan 50 instancias aleatoriamente. Para cada una de ellas, se resuelve el modelo de juego de seguridad mediante el uso del algoritmo de múltiples problemas lineales, resolviendo para cada estrategia pura del atacante un problema maestro y utilizando la heurística para resolver Dec-MDP elegida en cada uno de sus subproblemas. Sea κ el promedio de las columnas generadas en las 50 instancias mencionadas. El valor de este parámetro corresponde a una aproximación para el promedio descrito.

También, se considera el tiempo promedio que se requiere para generar una nueva columna, mediante el uso de la heurística elegida, y el tiempo promedio para obtener una cota superior en los problemas maestros correspondientes a las instancias pertenecientes a cada uno de los conjuntos considerados. Como aproximación de estos parámetros, se utiliza los tiempos promedios presentados en las Tablas 4.5 y 4.6 . Sea λ y μ dichas aproximaciones para el tiempo promedio de la generación de una columna y obtención de una cota superior, respectivamente.

Así, el parámetro α se define como:

$$\alpha = \left\lceil \frac{\kappa\lambda}{\mu} \right\rceil, \quad (5.1)$$

mientras que el β :

$$\beta = \left\lceil \frac{\mu}{\lambda} \right\rceil. \quad (5.2)$$

El valor de α representa una aproximación del número de veces que calcular la cota superior requiere el mismo tiempo que resolver, mediante la heurística elegida, todos los subproblemas asociados a los problemas maestros para una estrategia pura del atacante. Por otro lado, el valor de β corresponde a una aproximación del número de columnas cuyos Dec-MDP asociados requieren en total para su resolución, mediante dicha heurística, el mismo tiempo que el cálculo de la cota superior.

La Tabla 5.1 presenta el valor de los parámetros descritos para los conjuntos de instancias utilizados.

	Conjunto Nº1	Conjunto Nº2	Conjunto Nº3	Conjunto Nº4	Conjunto Nº5	Conjunto Nº6
κ	24	127	135	274	255	447
λ	0.36686	2.0845	3.7902	12.326	12.332	29.669
μ	6.0613	58.033	178.52	930.51	1272.9	27869
α	1	4	2	3	2	1
β	17	28	48	76	104	940

Tabla 5.1: Valor de κ , λ , μ , α y β para cada conjunto de instancias considerado.

5.1.2. Enfoque Basado en Cota Superior Probabilística

Finalmente, el segundo de los enfoques mencionados consiste en aproximar la cota superior del valor óptimo de un Dec-MDP, y utilizar dicha aproximación para obtener una cota superior calculable del valor óptimo para los problemas maestros en cuestión. Para esto, se hace uso de los experimentos desarrollados en la Sección 4.2.2. En específico, se utiliza la razón entre una solución factible de los Dec-MDP considerados y su respectiva cota superior calculada. La solución factible

utilizada es la otorgada por la heurística seleccionada, la que corresponde a Warm Start. Para cada conjunto de heurísticas considerado, se genera la distribución acumulada de dicha razón utilizando los datos empíricos obtenidos en los experimentos mencionados. La Tabla 5.2 presenta los valores para la media y desviación estándar para dicha razón en cada uno de los conjuntos de instancias considerados.

	Media	Desviación Estándar
Conjunto Nº1	0.9309	0.0391
Conjunto Nº2	0.9317	0.0333
Conjunto Nº3	0.9338	0.0317
Conjunto Nº4	0.9341	0.0297
Conjunto Nº5	0.9260	0.0325
Conjunto Nº6	0.9271	0.0307

Tabla 5.2: Media y desviación estándar de la razón entre la solución heurística y su cota superior para cada conjunto de instancias considerado.

Se observa que la heurística Warm Start otorga, en promedio, garantías superiores al 90 % para todos los conjuntos de instancias considerados. Esto permite evaluar la calidad de la solución subóptima obtenida mediante dicha heurística.

Sea v el valor de la solución heurística y η el valor de la cota superior para un Dec-MDP asociado a una de las instancias para un determinado conjunto. Sea el parámetro ω , el cual corresponde al mínimo valor de la razón descrita que asegura con probabilidad no menor a σ que el valor para dicha razón es mayor a él para las instancias del conjunto considerado. Mediante el uso de la distribución acumulada mencionada es posible obtener dicho parámetro, el cual corresponde a:

$$\mathbb{P}\left(\frac{v}{\eta} \geq \omega\right) \geq \sigma \quad (5.3)$$

Con esto, empíricamente se tiene que la desigualdad:

$$\frac{v}{\omega} \geq \eta \quad (5.4)$$

se cumple con probabilidad no menor a σ . Esta desigualdad puede ser utilizada en la ecuación (3.32) para obtener la siguiente aproximación de la cota superior para un problema maestro:

$$v^* \leq v + \frac{v}{\omega} - z; \quad (5.5)$$

en donde, z corresponde a la variable dual asociada a la restricción de igualdad del problema maestro que actualmente se está resolviendo, y v la solución heurística de su subproblema.

Este segundo enfoque consiste en emplear dicha aproximación para la cota superior del subproblema cada vez que una nueva columna es generada, con el propósito de descartar problemas

maestros cuya solución final no es de utilidad. Si bien, este enfoque no garantiza que las cotas superiores calculadas lo sean en realidad, se provee de una aproximación de ellas cuyo tiempo de obtención es despreciable.

La Tabla 5.3 presenta los valores para el parámetro ω para cada uno de los conjuntos de instancias considerados.

	Conjunto Nº1	Conjunto Nº2	Conjunto Nº3	Conjunto Nº4	Conjunto Nº5	Conjunto Nº6
ω	0.87496	0.88228	0.88687	0.88985	0.87802	0.8814

Tabla 5.3: Valor de ω para cada conjunto de instancias considerado.

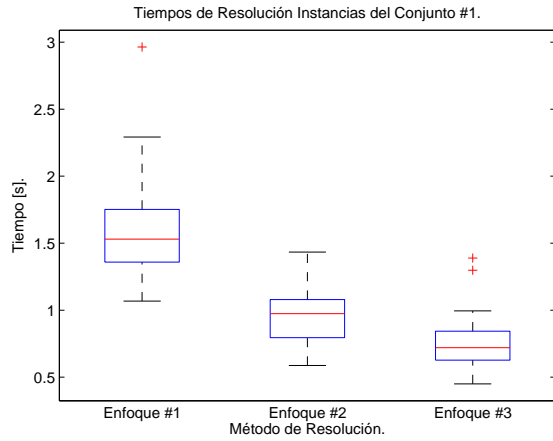
5.2. Experimentos Aleatorios

Con el propósito de evaluar los enfoques de resolución aquí presentados, se generan 50 instancias aleatorias para cada uno de los seis conjuntos de instancias considerados. Las utilidades para cada una de dichas instancias son generadas según el proceso descrito anteriormente en el presente capítulo. Cada una de estas instancias es resuelta mediante los tres enfoques presentados; siendo el enfoque #1 el correspondiente a la generación de todas las columnas, el enfoque #2 el correspondiente a la utilización de un número restringido de cotas superiores, y el enfoque #3 el correspondiente a la utilización de una cota superior probabilística. Para el caso específico de este trabajo, se hace uso de la cota superior del valor óptimo para el problema maestro descrita en la ecuación (4.46), la cual se obtiene mediante la resolución de la relajación lineal de un MILP.

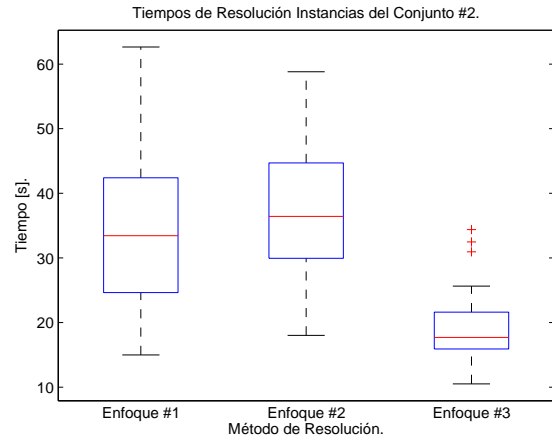
Para cada una de las instancias pertenecientes a los cinco primeros conjuntos utilizados, el resultado obtenido por cada uno de los enfoques de resolución fue el mismo. Sin embargo, las instancias del conjunto Nº6 no pudieron ser resueltas, ya que se tarda más del tiempo determinado como prudente, correspondiente a 12 horas, para la obtención de sus respectivos resultados. Debido a esto, las instancias pertenecientes a dicho conjunto son descartadas.

La Figura 5.1 presenta los *box plots* correspondiente a los tiempos de resolución, mediante los tres enfoques utilizados, para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.

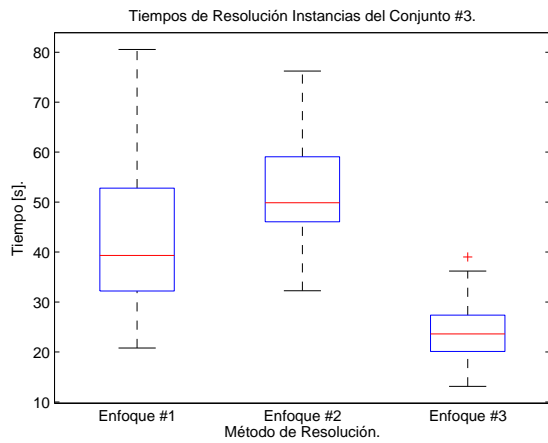
Para los cinco conjuntos de instancias utilizados, el enfoque #3 presenta la menor mediana para el tiempo de resolución. Mientras que para el conjunto Nº1, Nº4 y Nº5, el enfoque #1 presenta la mayor mediana para el tiempo de resolución. Para los conjuntos Nº2 y Nº3, la mayor mediana del tiempo de resolución corresponde a la utilización del enfoque #2. Además, el enfoque #3 presenta el menor rango de variación para su tiempo de resolución. Se aprecia que los tres enfoques de resolución utilizados presentan *outliers* para sus respectivos tiempos de resolución, considerando las distribuciones para dicho tiempo obtenidas mediante el uso de *box plots*.



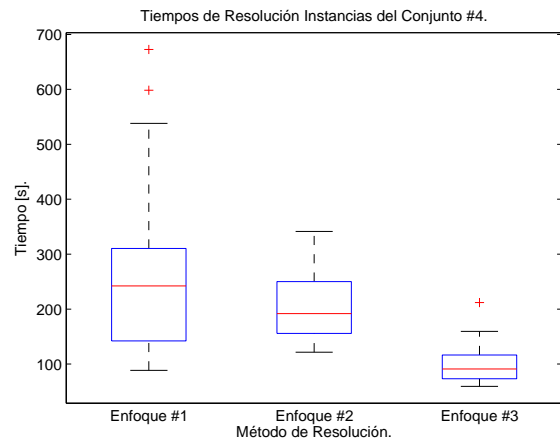
(a) Tiempos para instancias del conjunto Nº1.



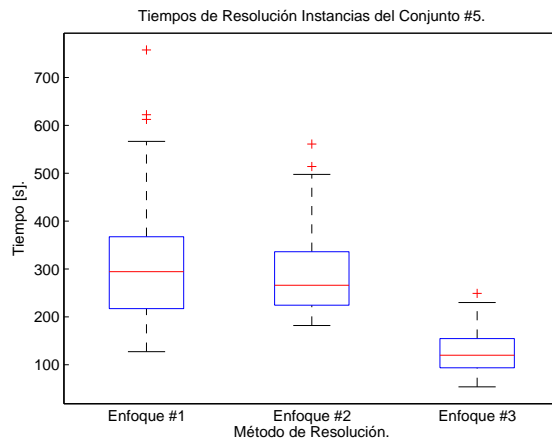
(b) Tiempos para instancias del conjunto Nº2.



(c) Tiempos para instancias del conjunto Nº3.



(d) Tiempos para instancias del conjunto Nº4.



(e) Tiempos para instancias del conjunto Nº5.

Figura 5.1: Tiempos de resolución utilizando diversos enfoques para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.

La Tabla 5.4 resume las estadísticas del tiempo de resolución, para los conjuntos de instancias

considerados, utilizando los enfoques #1, #2 y #3.

	Enf. # 1 Media [s]	Enf. # 1 Desv. Est. [s]	Enf. # 2 Media [s]	Enf. # 2 Desv. Est. [s]	Enf. # 3 Media [s]	Enf. # 3 Desv. Est. [s]
Conjunto Nº1	1.6043	0.3325	0.9558	0.2049	0.7440	0.1905
Conjunto Nº2	34.1819	11.3141	37.3473	9.6027	18.9164	5.0772
Conjunto Nº3	43.6268	15.3381	52.0296	9.7528	24.0929	5.5810
Conjunto Nº4	258.5533	132.0757	202.5337	53.9656	100.6651	31.7595
Conjunto Nº5	321.9592	138.3634	287.7973	87.9250	129.3801	47.4950

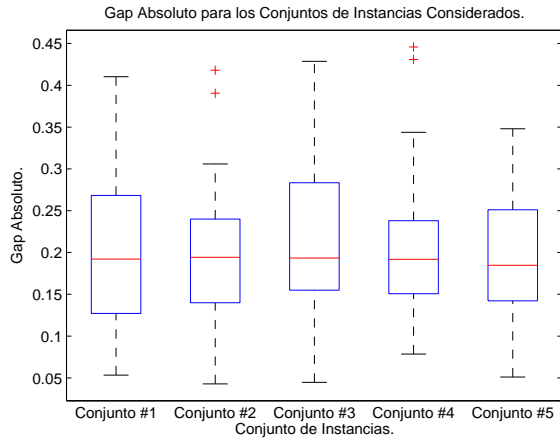
Tabla 5.4: Tiempos de resolución al utilizar los enfoques #1, #2 y #3 para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.

Se aprecia que el enfoque #3 presenta el menor tiempo promedio para todos los conjuntos de instancias considerados. Para el enfoque #2, sus tiempos promedios de resolución son mayores, en comparación los obtenidos por el enfoque #1, para los conjuntos de instancias Nº2 y Nº3. Para los conjuntos Nº1, Nº4 y Nº5, el enfoque #2 presenta menor tiempo promedio de resolución que el enfoque #1. Además, se obtiene la menor desviación estándar para el tiempo de resolución al utilizar el enfoque #3.

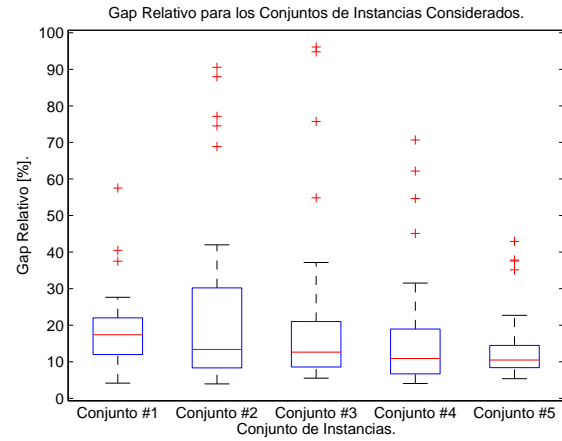
Para cada una de las instancias utilizadas se obtiene una garantía de su solución óptima. Para esto, se utiliza el enfoque de resolución # 1, calculando la cota superior a cada problema maestro al final de su respectivo proceso de generación de columnas. Estas cotas superiores son calculadas mediante la relajación lineal de la formulación MILP presentada. La garantía mencionada corresponde a la mayor de dichas cotas superiores. Esta garantía corresponde a una cota superior para el juego de seguridad correspondiente a cada una de las instancias consideradas. Los tiempos de resolución para los diversos enfoques presentados no consideran la obtención de esta garantía.

La Figura 5.2 presenta el gap absoluto y relativo para los conjuntos de instancias considerados mediante el uso de *box plots*. El gap absoluto, para un determinado conjunto de instancias, se obtiene haciendo la resta de la cota superior para cada una de las instancias pertenecientes a dicho conjunto y su respectiva mejor solución factible obtenida mediante alguno de los enfoques presentados. El gap relativo, para un determinado conjunto de instancias, se define haciendo la normalización del gap absoluto mediante la cota superior correspondiente. Sin embargo, para evitar el uso de un valor de referencia menor al valor que se pretende normalizar, sólo se consideran las instancias cuyo valor obtenido por los enfoques presentados es no negativo.

Se aprecia que para todos los conjuntos de instancias considerados, la mediana del valor correspondiente al gap absoluto pertenece al intervalo $[0.15; 0.2]$. Esta mediana es considerablemente pequeña en comparación al rango en el que se generan las utilidades para el defensor en dichas instancias, las cuales están en el intervalo $[0; 10]$. La mediana del valor para el gap relativo se encuentra entre el 10 % y 20 % para cada uno de los conjuntos de instancias considerados. Además, el rango en el que se encuentran las distribuciones para dicho gap es menor a 50 % para todos estos conjuntos. Sin embargo, los resultados obtenidos en cada uno de dichos conjuntos presentan *outliers*. Estos valores anómalos llegan a ser del orden del 90 % para el conjunto Nº3.



(a) Gap absoluto.



(b) Gap relativo.

Figura 5.2: Gap absoluto y relativo para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.

La Tabla 5.5 resume las estadísticas, para los conjuntos de instancias considerados, del gap absoluto y relativo.

	Gap Absoluto Media	Gap Absoluto Desv. Est.	Gap Relativo Media	Gap Relativo Desv. Est.
Conjunto Nº1	0.1983	0.0903	19.5820	11.5167
Conjunto Nº2	0.1914	0.0836	22.8909	22.5896
Conjunto Nº3	0.2120	0.0905	20.4943	21.0315
Conjunto Nº4	0.2051	0.0762	16.0820	14.9491
Conjunto Nº5	0.1940	0.0745	13.5993	8.8559

Tabla 5.5: Gap absoluto y relativo para los conjuntos de instancias Nº1, Nº2, Nº3, Nº4 y Nº5.

5.3. Ejemplo Representativo: Red de Metro

Con el objetivo de resolver un problema que asemeje un caso real, se ha diseñado una instancia inspiradas en la red correspondiente al Metro de Santiago, Chile. El grafo que caracteriza dicha instancia está basado en la conexión de las estaciones pertenecientes al centro de dicha ciudad. Las utilidades son generadas de igual forma que para las instancias resueltas anteriormente en este capítulo. El tamaño que describe a esta instancia corresponde al de las instancias pertenecientes al conjunto Nº6; es decir, 16 estaciones, 12 períodos de tiempo, y 6 recursos defensivos. Las conexiones entre estas estaciones, junto con el particionamiento para la utilización del método heurístico Warm Start, se presentan en la Figura 5.3.

Este grafo representa una red constituida por cuatro líneas de metro. Según la topología del metro de Santiago: la estación Los Heroes corresponde a la estación 2; las estaciones 1, 2, 3, 4, 5,

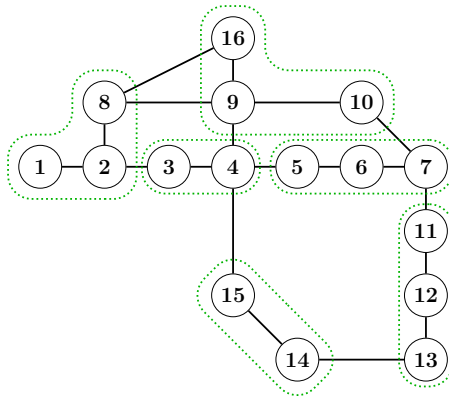


Figura 5.3: Grafo para la instancia de la red de metro.

6 y 7 pertenecen a la línea #1; las estaciones 2, 8 y 16 a la línea #2; las estaciones 8, 9, 10, 7, 11, 12 y 13 a la línea #5; y las estaciones 13, 14, 15, 4, 9 y 16 a la línea #3.

Debido al tamaño de esta instancia, su resolución sólo se realiza mediante el enfoque #3. Para evaluar el desempeño de dicho enfoque en el contexto presentado, se generan 50 de las instancias descritas de forma aleatoria.

La Figura 5.4 presenta el *box plot* para el tiempo de resolución al utilizar el enfoque #3 en instancias correspondientes a la red de metro.

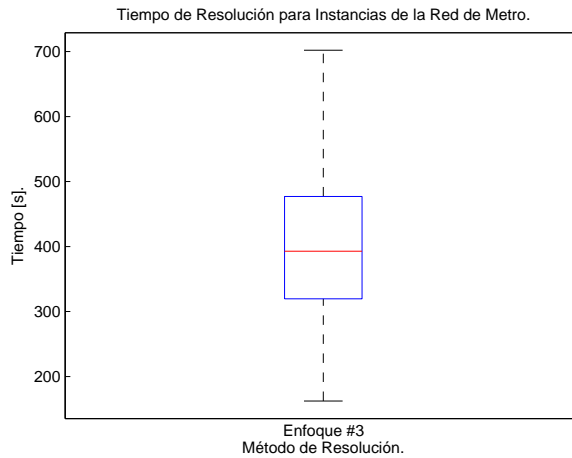


Figura 5.4: Tiempo de resolución mediante el enfoque #3 para instancias de la red de metro.

Se aprecia que la mediana para el tiempo de resolución es de aproximadamente 400 [s], y que el rango en el que se encuentra la distribución para dicho tiempo queda determinado, aproximadamente, entre 200 y 700 segundos.

La Tabla 5.6 resume las estadísticas del tiempo de resolución, para las instancias correspondientes a la red de metro, utilizando el enfoque #3.

	Media [s]	Desv. Est. [s]
Red de Metro	412.2333	133.5756

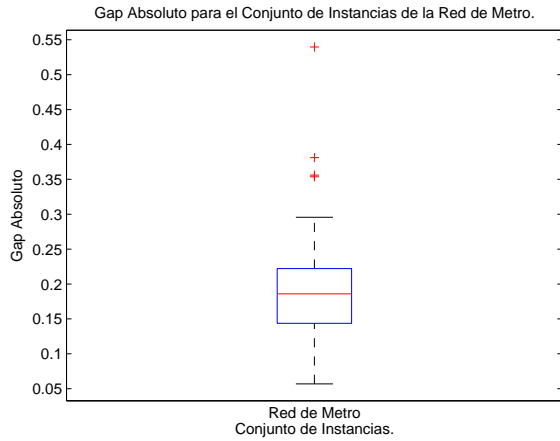
Tabla 5.6: Tiempo de resolución mediante el enfoque #3 para instancias de la red de metro.

Se aprecia que el valor promedio del tiempo de resolución se aproxima a la mediana de su valor. También, el valor de su desviación estándar se aproxima al rango en el que la distribución para el tiempo de resolución, presentado en el *box plot*, posee el 25% de muestras superiores e inferiores a su respectiva mediana.

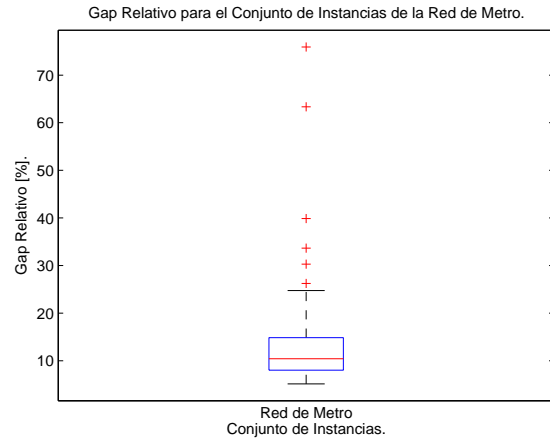
Para las instancias de la red de metro presentadas, se calcula una garantía para las soluciones obtenidas mediante el enfoque utilizado. Esta garantía es calculada de igual forma que para las instancias aleatorias resueltas en la sección anterior. Los tiempos de resolución presentados para dicho enfoque no consideran el cálculo esta garantía para cada instancia resuelta.

La Figura 5.5 presenta el gap absoluto y relativo para las instancias correspondientes a la red de metro.

Se aprecia que la distribución para el gap absoluto tiene una mediana de aproximadamente 0.2,



(a) Gap absoluto.



(b) Gap relativo.

Figura 5.5: Gap absoluto y relativo para instancias de la red de metro.

y que su rango en donde está definida corresponde, aproximadamente, a $[0.05; 0.3]$. Existe la presencia de valores anómalos para su distribución, los cuales llegan al orden de 0.55. La distribución para el gap relativo presenta un rango definido, aproximadamente, en el intervalo comprendido entre 5% y 25%; y cuya mediana corresponde, aproximadamente, a 10%. Se aprecia la presencia de valores anómalos, los cuales llegan al orden de 70%.

La Tabla 5.7 resume las estadísticas del gap absoluto y relativo para las instancias correspondientes a la red de metro.

	Gap Absoluto Media	Gap Absoluto Desv. Est.	Gap Relativo Media	Gap Relativo Desv. Est.
Red de Metro	0.2007	0.0861	15.0279	13.7742

Tabla 5.7: Gap absoluto y relativo para instancias de la red de metro.

5.3.1. Visualización de Políticas

Una forma de visualizar una política conjunta que utilizan los agentes es mediante el uso de grafos acíclicos dirigidos. Para cada uno de los agentes se utiliza uno de dichos grafos para representar su política local. A modo de ejemplo se muestra en la Figura 5.6 la política que debe ejecutar un agente en la instancia de la red de metro. La ubicación espacial horizontal de los nodos indica la estación en la que se encuentra el agente, mientras que la ubicación vertical el período de tiempo. El número dentro de cada nodo indica la acción que debe tomar el agente que se encuentra en dicho lugar y período. Las arcos dirigidos en color negro indican el éxito en la realización de la acción realizada por el agente; mientras que los arcos dirigidos en color rojo y punteados indican su fracaso, dejando al agente en la misma estación al siguiente período de tiempo. Así, según el ejemplo presentado, el agente comienza encontrándose en la estación 4, en donde se queda hasta

el período 3. En este período, realiza la acción de moverse a la estación 3. Como resultado de esta acción, el agente puede encontrarse en la estación 3 o permanecer en la estación 4 en el período 4. Sucesivamente, el grafo presenta las acciones que debe tomar el agente en cada una de las estaciones y períodos de tiempo en que se encuentre.

En este ejemplo se aprecia que el agente sólo toma acciones que le permita visitar las estaciones 1, 2, 3 y 4. Esto es una consecuencia del enfoque de resolución greedy. Tomando en cuenta las políticas de los agentes previamente resueltos, la mayor utilidad marginal para el sistema se logra al concentrar un recurso en dichas estaciones. En los primeros períodos de tiempo el agente toma la acción de permanecer en la estación 4. Luego intenta moverse hacia la estación 1. Finalmente, en caso de encontrarse en las estaciones 3 o 4 toma acciones que lo lleven a la estación 4, y en caso de encontrarse en las estaciones 1 o 2 toma acciones que lo lleven a la estación 1.

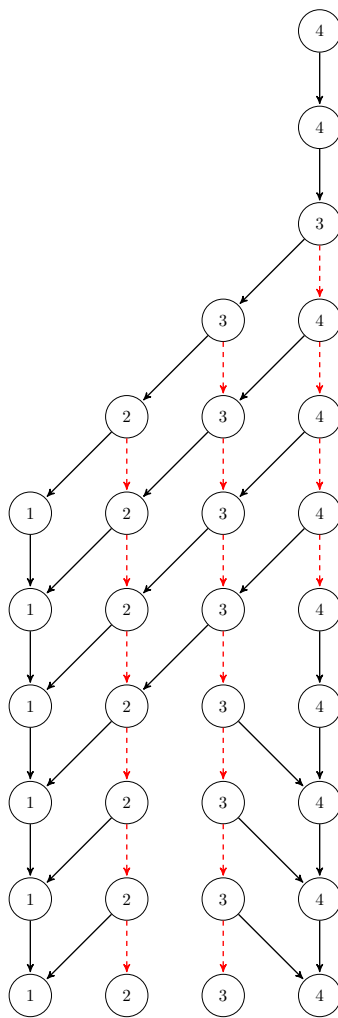


Figura 5.6: Visualización de la política empleada por un agente.

Capítulo 6

Conclusiones

En este trabajo se presentan tres enfoques de resolución para un juego de seguridad de Stackelberg para la optimización de la estrategia de múltiples recursos defensores descentralizados en presencia de incertidumbre. Estos enfoques hacen uso de una particularidad del esquema de resolución del modelo, basado en generación de columnas. Estos enfoques otorgan una solución factible subóptima para dicho problema, ya que están basados en la utilización de métodos heurísticos. Para todas las instancias de prueba, diseñadas para la evaluación empírica de dichos enfoques de resolución, se obtiene que los tres otorgan el mismo resultado. El enfoque # 2 presenta tiempos de resolución similares al enfoque original, siendo menor para algunos de los conjuntos de instancias considerados. Sin embargo, el enfoque # 3 presenta un tiempo de resolución menor al requerido por el enfoque original para todos los conjuntos de instancias considerados.

Además, se presenta una formulación para calcular una cota superior del valor de los problemas maestros correspondientes al modelo de juego de seguridad. Esta formulación no es factible de ser resuelta, ya que contempla la resolución exacta de un problema Dec-MDP. Sin embargo, se desarrolla una forma novedosa de calcular dicha cota superior. La obtención de dicha cota permite desarrollar los enfoques de resolución para el juego de seguridad presentados. También, permite obtener garantías de las soluciones subóptimas entregadas por dichos enfoques de resolución. Estas garantías corresponden a los gap calculados en las instancias de prueba. Para el gap relativo, se tiene en promedio valores en torno al 20% para cada conjunto de instancias considerado. Además el rango para la distribución de dicho gap en cada uno de estos conjuntos no supera el 50%. Por otro lado, el gap absoluto, para cada uno de estos conjuntos de instancias, tiene un valor medio de 0.2, siendo su valor máximo para su respectiva distribución en torno a 0.4. Se observa empíricamente que tanto las distribuciones para el gap relativo y absoluto no tienen grandes alteraciones para dichos conjuntos de instancias.

El enfoque de resolución #3 permite resolver una instancia inspirada en una red de metro real. Esta instancia es suficientemente grande como para representar un problema de seguridad real. El enfoque utilizado otorga soluciones que presentan un gap absoluto en torno a 0.2 y un gap relativo en torno a 10%. El tiempo de resolución para obtener la solución aproximada está en torno a 400 segundos. Esto hace posible la solución de instancias de dicho tamaño, pero sin verificar la calidad

de la solución inmediatamente.

Por último, se presenta la comparación de diversas heurísticas para la resolución de la problemática Dec-MDP. Algunas de dichas heurísticas fueron desarrolladas en este trabajo, como lo son la heurística basada en una formulación MILP y los métodos Warm Start y Cold Start. Además, se presenta una metodología para la elección de los parámetros de la heurística Entropía Cruzada. Para la evaluación del desempeño de las heurísticas presentadas se hizo uso de simulación de instancias aleatorias. Esto permite la elección de la heurística más provechosa para ser utilizada para resolver el subproblema de los enfoques de resolución para el juego de seguridad presentados. Para el caso de este trabajo, dicha heurística corresponde a Warm Start. También, es posible obtener garantías para las heurísticas presentadas mediante el cálculo de una cota superior de su correspondiente valor óptimo. Para el caso de la heurística Warm Start, se obtuvo en promedio garantías de más del 90 %. Estas heurísticas, junto con la obtención de su garantía, pueden ser utilizadas para problemas Dec-MDP en general, no siendo su uso restringido únicamente para el caso de estudio de este trabajo.

Los enfoques de solución desarrollados en este trabajo fueron evaluados en un juego de seguridad inspirado en una red de metro. Sin embargo, estos enfoques son utilizables para juegos de seguridad en general, permitiendo la obtención de itinerarios eficientes para la utilización de múltiples recursos en la seguridad de aeropuertos, puertos, fronteras entre países, áreas urbanas, etc. Más aún, el uso de estos enfoques no se limita sólo a juegos de seguridad, ya que son posibles de ser utilizados en modelos que admitan solución mediante el algoritmo de múltiples problemas lineales, cuyos problemas asociados permitan su resolución utilizando un enfoque de generación de columnas.

Bibliografía

- [1] Zhengyu Yin, Albert Xin Jiang, Matthew Paul Johnson, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, Milind Tambe, and John P Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012.
- [2] William B Haskell, Debarun Kar, Fei Fang, Milind Tambe, Sam Cheung, and Elizabeth De-nicola. Robust protection of fisheries with compass. In *AAAI*, pages 2978–2983, 2014.
- [3] Rong Yang, Benjamin Ford, Milind Tambe, and Andrew Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 453–460. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [4] James Pita, Manish Jain, Janusz Marecki, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, pages 125–132. International Foundation for Autonomous Agents and Multiagent Systems, 2008.
- [5] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. Protect: A deployed game theoretic system to protect the ports of the united states. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 13–20. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [6] Yundi Qian, William B Haskell, Albert Xin Jiang, and Milind Tambe. Online planning for optimal protector strategies in resource conservation games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 733–740. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [7] Victor Bucarey, Carlos Casorrán, Óscar Figueroa, Karla Rosas, Hugo Navarrete, and Fernando Ordóñez. Building real stackelberg security games for border patrols. In *International Conference on Decision and Game Theory for Security*, pages 193–212. Springer, 2017.
- [8] Heinrich von Stackelberg et al. Theory of the market economy. 1952.
- [9] Jerome Bracken and James T McGill. Mathematical programs with optimization problems in

- the constraints. *Operations Research*, 21(1):37–44, 1973.
- [10] George Leitmann. On generalized stackelberg strategies. *Journal of optimization theory and applications*, 26(4):637–643, 1978.
- [11] Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 895–902. International Foundation for Autonomous Agents and Multiagent Systems, 2008.
- [12] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 689–696. International Foundation for Autonomous Agents and Multiagent Systems, 2009.
- [13] Rong Yang, Fernando Ordonez, and Milind Tambe. Computing optimal strategy against quantal response in security games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 847–854. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [14] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90. ACM, 2006.
- [15] Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.
- [16] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 327–334. International Foundation for Autonomous Agents and Multiagent Systems, 2011.
- [17] Manish Jain, Jason Tsai, James Pita, Christopher Kiekintveld, Shyamsunder Rathi, Milind Tambe, and Fernando Ordóñez. Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40(4):267–290, 2010.
- [18] Albert Xin Jiang, Zhengyu Yin, Chao Zhang, Milind Tambe, and Sarit Kraus. Game-theoretic randomization for security patrolling with dynamic execution uncertainty. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 207–214. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [19] James Pita, Manish Jain, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.
- [20] Eric Anyung Shieh, Manish Jain, Albert Xin Jiang, and Milind Tambe. Efficiently solving joint activity based security games. In *IJCAI*, pages 346–352, 2013.

- [21] Zhengyu Yin, Dmytro Korzhyk, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. nash in security games: Interchangeability, equivalence, and uniqueness. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1139–1146. International Foundation for Autonomous Agents and Multiagent Systems, 2010.
- [22] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [23] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210. Morgan Kaufmann Publishers Inc., 1996.
- [24] Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V Goldman. Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, 22:423–455, 2004.
- [25] Richard Washington, Keith Golden, John Bresina, David E Smith, Corin Anderson, and Trey Smith. Autonomous rovers for mars exploration. In *Aerospace Conference, 1999. Proceedings. 1999 IEEE*, volume 1, pages 237–251. IEEE, 1999.
- [26] Sameera S Ponda, Luke B Johnson, Alborz Geramifard, and Jonathan P How. Cooperative mission planning for multi-uav teams. In *Handbook of Unmanned Aerial Vehicles*, pages 1447–1490. Springer, 2015.
- [27] Jilles S Dibangoye, Christopher Amato, and Arnaud Doniec. Scaling up decentralized mdps through heuristic search. *arXiv preprint arXiv:1210.4865*, 2012.
- [28] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.
- [29] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.
- [30] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
- [31] George Dantzig. *Linear programming and extensions*. Princeton university press, 2016.
- [32] Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.
- [33] Eric Shieh, Albert Jiang, Amulya Yadav, Pradeep Reddy VARAKANTHAM, and Milind Tambe. Unleashing dec-mdps in security games: Enabling effective defender teamwork. 2014.
- [34] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.

- [35] Pradeep Varakantham, Hoong Chuin Lau, and Zhi Yuan. Scalable randomized patrolling for securing rapid transit networks. In *IAAI*, 2013.
- [36] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184:78–123, 2012.
- [37] Daniel S Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 32–37. Morgan Kaufmann Publishers Inc., 2000.
- [38] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.
- [39] G Alon, Dirk P Kroese, Tal Raviv, and Reuven Y Rubinstein. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, 134(1):137–151, 2005.
- [40] Zdravko Botev and Dirk P Kroese. Global likelihood optimization via the cross-entropy method with an application to mixture models. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 1. IEEE, 2004.
- [41] Izack Cohen, Boaz Golany, and Avraham Shtub. Managing stochastic, finite capacity, multi-project systems through the cross-entropy methodology. *Annals of Operations Research*, 134(1):183–199, 2005.
- [42] Shie Mannor, Reuven Y Rubinstein, and Yohai Gat. The cross entropy method for fast policy search. In *ICML*, pages 512–519, 2003.
- [43] Rosemary Emery-Montemerlo, Geoff Gordon, Jeff Schneider, and Sebastian Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 136–143. IEEE Computer Society, 2004.
- [44] Sven Seuken and Shlomo Zilberstein. Improved memory-bounded dynamic programming for decentralized pomdps. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI'07)*, pages 344–351, 2007.
- [45] Sven Seuken and Shlomo Zilberstein. Memory-bounded dynamic programming for dec-pomdps. In *IJCAI*, pages 2009–2015, 2007.
- [46] Daniel Szer and François Charpillet. Point-based dynamic programming for dec-pomdps. In *AAAI*, volume 6, pages 1233–1238, 2006.
- [47] Frans A Oliehoek, Julian FP Kooij, Nikos Vlassis, et al. The cross-entropy method for policy search in decentralized pomdps. *Informatica*, 32(4):341–357, 2008.
- [48] Ranjit Nair, Milind Tambe, Makoto Yokoo, David Pynadath, and Stacy Marsella. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *IJCAI*,

pages 705–711, 2003.

[49] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

Anexos

Elección de Parámetros para Heurística Entropía Cruzada

Esta metodología consiste en, para cada uno de los conjuntos de instancias presentados en el Capítulo 4, generar 10000 instancias aleatorias por cada una de las cuatro distribuciones para las utilidades descritas en dicho capítulo. Cada una de estas instancias es resuelta mediante la heurística Entropía Cruzada, utilizando un conjunto de valores para el parámetro α . Este conjunto corresponde a $\alpha = \{0.1; 0.2; 0.3; \dots; 0.9\}$, junto con valores iniciales para el resto de los parámetros. Los valores promedios de los resultados obtenidos, para cada uno de los valores del parámetro α considerados, son comparados entre sí mediante intervalos de confianza. Para esto, se hace la resta de los resultados obtenidos con el uso de dos valores distintos para dicho parámetro y se calcula el intervalo de confianza para determinar si el uso de uno de esos valores es estadísticamente significativo en comparación al otro.

Para el conjunto Nº1, el valor inicial de los parámetros corresponde a: $I = 50$, $N = 50$ y $\rho = 0.1$. La Tabla 6.1 presenta los intervalos de confianza para el parámetro α obtenidos al resolver instancias pertenecientes a dicho conjunto.

El valor del parámetro α para el conjunto Nº1 que presenta intervalos de confianza positivos, remarcado con color, corresponde a 0.2. Este valor es el elegido para dicho parámetro.

De forma análoga, se considera para el parámetro I el siguiente conjunto de valores: $I = \{10; 20; 30; \dots; 180\}$. Las Tablas 6.2, 6.3 y 6.4 presentan los intervalos de confianza para dicho parámetro.

El valor elegido para el parámetro I corresponde a 90, y sus intervalos de confianza se remarcan en color.

Para los valores del parámetro N se considera el conjunto $N = \{10; 20; 30; \dots; 110\}$. Los intervalos de confianza para los valores de dicho parámetro se presentan en las Tablas 6.5 y 6.6.

Se aprecia en las Tablas 6.5 y 6.6 que a medida que el valor para el parámetro N aumenta se obtienen resultados estadísticamente significativos en comparación a valores menores para dicho parámetro. Sin embargo, dicho aumento disminuye marginalmente a medida que se utiliza un valor mayor para el parámetro. Esto lleva a que la obtención de este parámetro se realice mediante un

Parámetro α	0.1	0.2	0.3
0.1	[0; 0]	[0.0226; 0.0277]	[-0.0859; -0.0787]
0.2	[-0.0277; -0.0226]	[0; 0]	[-0.1111; -0.1037]
0.3	[0.0787; 0.0859]	[0.1037; 0.1111]	[0; 0]
0.4	[0.2130; 0.2226]	[0.2380; 0.2478]	[0.1302; 0.1407]
0.5	[0.3539; 0.3662]	[0.3789; 0.3914]	[0.2714; 0.2841]
0.6	[0.4990; 0.5138]	[0.5240; 0.5390]	[0.4166; 0.4316]
0.7	[0.6445; 0.6618]	[0.6696; 0.6870]	[0.5622; 0.5794]
0.8	[0.7891; 0.8089]	[0.8142; 0.8341]	[0.7069; 0.7265]
0.9	[0.9482; 0.9706]	[0.9733; 0.9958]	[0.8661; 0.8881]
	0.4	0.5	0.6
0.1	[-0.2226; -0.2130]	[-0.3662; -0.3539]	[-0.5138; -0.4990]
0.2	[-0.2478; -0.2380]	[-0.3914; -0.3789]	[-0.5390; -0.5240]
0.3	[-0.1407; -0.1302]	[-0.2841; -0.2714]	[-0.4316; -0.4166]
0.4	[0; 0]	[-0.1490; -0.1356]	[-0.2962; -0.2810]
0.5	[0.1356; 0.1490]	[0; 0]	[-0.1543; -0.1384]
0.6	[0.2810; 0.2962]	[0.1384; 0.1543]	[0; 0]
0.7	[0.4268; 0.4440]	[0.2843; 0.3018]	[0.1377; 0.1558]
0.8	[0.5716; 0.5909]	[0.4293; 0.4486]	[0.2827; 0.3025]
0.9	[0.7309; 0.7524]	[0.5886; 0.6101]	[0.4423; 0.4637]
	0.7	0.8	0.9
0.1	[-0.6618; -0.6445]	[-0.8089; -0.7891]	[-0.9706; -0.9482]
0.2	[-0.6870; -0.6696]	[-0.8341; -0.8142]	[-0.9958; -0.9733]
0.3	[-0.5794; -0.5622]	[-0.7265; -0.7069]	[-0.8881; -0.8661]
0.4	[-0.4440; -0.4268]	[-0.5909; -0.5716]	[-0.7524; -0.7309]
0.5	[-0.3018; -0.2843]	[-0.4486; -0.4293]	[-0.6101; -0.5886]
0.6	[-0.1558; -0.1377]	[-0.3025; -0.2827]	[-0.4637; -0.4423]
0.7	[0; 0]	[-0.1559; -0.1358]	[-0.3170; -0.2955]
0.8	[0.1358; 0.1559]	[0; 0]	[-0.1716; -0.1492]
0.9	[0.2955; 0.3170]	[0.1492; 0.1716]	[0; 0]

Tabla 6.1: Ajuste parámetro α para conjunto N°1.

proceso alternativo. Este proceso consiste en graficar el aumento porcentual del promedio de los resultados obtenidos mediante valores consecutivos para el parámetro N , con respecto al promedio obtenido mediante el mayor valor utilizado para dicho parámetro. El valor elegido para el parámetro N corresponde al menor valor que garantice un aumento marginal mayor a 70%. La Figura 6.1 presenta dicho proceso alternativo.

Así, el valor elegido para el parámetro N corresponde a 50.

Para el parámetro ρ se considera el conjunto de valores $\rho = \{0.1; 0.2; 0.3; \dots; 0.9\}$. La Tabla 6.7 presenta los intervalos de confianza para dicho parámetro.

El valor elegido para el parámetro ρ corresponde a 0.3, cuyos intervalos de confianza se remarcan

Parámetro I	10	20	30	40
10	[0; 0]	[0.6805; 0.6982]	[0.8795; 0.8986]	[0.9415; 0.9609]
20	[-0.6982; -0.6805]	[0; 0]	[0.1952; 0.2041]	[0.2573; 0.2663]
30	[-0.8986; -0.8795]	[-0.2041; -0.1952]	[0; 0]	[0.0590; 0.0653]
40	[-0.9609; -0.9415]	[-0.2663; -0.2573]	[-0.0653; -0.0590]	[0; 0]
50	[-0.9827; -0.9631]	[-0.2880; -0.2790]	[-0.0870; -0.0807]	[-0.0245; -0.0188]
60	[-0.9915; -0.9719]	[-0.2969; -0.2878]	[-0.0958; -0.0895]	[-0.0334; -0.0277]
70	[-0.9962; -0.9765]	[-0.3015; -0.2924]	[-0.1004; -0.0942]	[-0.0380; -0.0323]
80	[-0.9982; -0.9784]	[-0.3035; -0.2944]	[-0.1024; -0.0961]	[-0.0399; -0.0343]
90	[-0.9989; -0.9792]	[-0.3043; -0.2951]	[-0.1032; -0.0969]	[-0.0407; -0.0350]
100	[-0.9999; -0.9801]	[-0.3052; -0.2961]	[-0.1041; -0.0979]	[-0.0416; -0.0360]
110	[-1.0002; -0.9805]	[-0.3055; -0.2964]	[-0.1045; -0.0982]	[-0.0420; -0.0363]
120	[-0.9991; -0.9794]	[-0.3044; -0.2953]	[-0.1034; -0.0971]	[-0.0409; -0.0352]
130	[-1.0016; -0.9819]	[-0.3070; -0.2978]	[-0.1058; -0.0996]	[-0.0434; -0.0377]
140	[-1.0011; -0.9813]	[-0.3064; -0.2973]	[-0.1053; -0.0990]	[-0.0428; -0.0372]
150	[-0.9991; -0.9794]	[-0.3044; -0.2953]	[-0.1033; -0.0971]	[-0.0409; -0.0352]
160	[-1.0014; -0.9816]	[-0.3067; -0.2975]	[-0.1056; -0.0994]	[-0.0431; -0.0375]
170	[-1.0011; -0.9813]	[-0.3064; -0.2973]	[-0.1053; -0.0990]	[-0.0428; -0.0372]
180	[-0.9999; -0.9802]	[-0.3052; -0.2961]	[-0.1041; -0.0978]	[-0.0416; -0.0360]
	50	60	70	80
10	[0.9631; 0.9827]	[0.9719; 0.9915]	[0.9765; 0.9962]	[0.9784; 0.9982]
20	[0.2790; 0.2880]	[0.2878; 0.2969]	[0.2924; 0.3015]	[0.2944; 0.3035]
30	[0.0807; 0.0870]	[0.0895; 0.0958]	[0.0942; 0.1004]	[0.0961; 0.1024]
40	[0.0188; 0.0245]	[0.0277; 0.0334]	[0.0323; 0.0380]	[0.0343; 0.0399]
50	[0; 0]	[0.0061; 0.0117]	[0.0107; 0.0162]	[0.0127; 0.0182]
60	[-0.0117; -0.0061]	[0; 0]	[0.0019; 0.0074]	[0.0038; 0.0093]
70	[-0.0162; -0.0107]	[-0.0074; -0.0019]	[0; 0]	[-0.0008; 0.0047]
80	[-0.0182; -0.0127]	[-0.0093; -0.0038]	[-0.0047; 0.0008]	[0; 0]
90	[-0.0190; -0.0134]	[-0.0101; -0.0046]	[-0.0054; 0.0000]	[-0.0035; 0.0020]
100	[-0.0199; -0.0144]	[-0.0110; -0.0056]	[-0.0064; -0.0009]	[-0.0044; 0.0010]
110	[-0.0202; -0.0147]	[-0.0114; -0.0059]	[-0.0067; -0.0013]	[-0.0048; 0.0007]
120	[-0.0192; -0.0136]	[-0.0103; -0.0048]	[-0.0056; -0.0002]	[-0.0037; 0.0018]
130	[-0.0216; -0.0162]	[-0.0127; -0.0073]	[-0.0081; -0.0027]	[-0.0061; -0.0007]
140	[-0.0211; -0.0156]	[-0.0122; -0.0068]	[-0.0076; -0.0022]	[-0.0056; -0.0002]
150	[-0.0191; -0.0136]	[-0.0102; -0.0048]	[-0.0056; -0.0002]	[-0.0037; 0.0018]
160	[-0.0214; -0.0159]	[-0.0125; -0.0071]	[-0.0078; -0.0025]	[-0.0059; -0.0005]
170	[-0.0211; -0.0156]	[-0.0122; -0.0067]	[-0.0076; -0.0021]	[-0.0056; -0.0002]
180	[-0.0199; -0.0144]	[-0.0111; -0.0055]	[-0.0064; -0.0010]	[-0.0044; 0.0010]

Tabla 6.2: Ajuste parámetro I para conjunto №1 (Tabla #1).

en color.

De esta forma, los parámetros de la heurística Entropía Cruzada, para las instancias pertenecientes al conjunto №1, corresponden a: $\alpha = 0.2$, $I = 90$, $N = 50$ y $\rho = 0.3$.

Parámetro I	90	100	110	120
10	[0.9792; 0.9989]	[0.9801; 0.9999]	[0.9805; 1.0002]	[0.9794; 0.9991]
20	[0.2951; 0.3043]	[0.2961; 0.3052]	[0.2964; 0.3055]	[0.2953; 0.3044]
30	[0.0969; 0.1032]	[0.0979; 0.1041]	[0.0982; 0.1045]	[0.0971; 0.1034]
40	[0.0350; 0.0407]	[0.0360; 0.0416]	[0.0363; 0.0420]	[0.0352; 0.0409]
50	[0.0134; 0.0190]	[0.0144; 0.0199]	[0.0147; 0.0202]	[0.0136; 0.0192]
60	[0.0046; 0.0101]	[0.0056; 0.0110]	[0.0059; 0.0114]	[0.0048; 0.0103]
70	[-0.0000; 0.0054]	[0.0009; 0.0064]	[0.0013; 0.0067]	[0.0002; 0.0056]
80	[-0.0020; 0.0035]	[-0.0010; 0.0044]	[-0.0007; 0.0048]	[-0.0018; 0.0037]
90	[0; 0]	[-0.0018; 0.0037]	[-0.0014; 0.0040]	[-0.0026; 0.0030]
100	[-0.0037; 0.0018]	[0; 0]	[-0.0024; 0.0030]	[-0.0035; 0.0020]
110	[-0.0040; 0.0014]	[-0.0030; 0.0024]	[0; 0]	[-0.0038; 0.0017]
120	[-0.0030; 0.0026]	[-0.0020; 0.0035]	[-0.0017; 0.0038]	[0; 0]
130	[-0.0054; 0.0001]	[-0.0044; 0.0009]	[-0.0041; 0.0013]	[-0.0052; 0.0002]
140	[-0.0049; 0.0005]	[-0.0039; 0.0015]	[-0.0036; 0.0018]	[-0.0047; 0.0007]
150	[-0.0029; 0.0026]	[-0.0019; 0.0035]	[-0.0016; 0.0038]	[-0.0027; 0.0028]
160	[-0.0051; 0.0003]	[-0.0042; 0.0012]	[-0.0038; 0.0015]	[-0.0050; 0.0005]
170	[-0.0049; 0.0006]	[-0.0039; 0.0015]	[-0.0036; 0.0019]	[-0.0047; 0.0008]
180	[-0.0037; 0.0018]	[-0.0027; 0.0027]	[-0.0024; 0.0031]	[-0.0035; 0.0020]
	130	140	150	160
10	[0.9819; 1.0016]	[0.9813; 1.0011]	[0.9794; 0.9991]	[0.9816; 1.0014]
20	[0.2978; 0.3070]	[0.2973; 0.3064]	[0.2953; 0.3044]	[0.2975; 0.3067]
30	[0.0996; 0.1058]	[0.0990; 0.1053]	[0.0971; 0.1033]	[0.0994; 0.1056]
40	[0.0377; 0.0434]	[0.0372; 0.0428]	[0.0352; 0.0409]	[0.0375; 0.0431]
50	[0.0162; 0.0216]	[0.0156; 0.0211]	[0.0136; 0.0191]	[0.0159; 0.0214]
60	[0.0073; 0.0127]	[0.0068; 0.0122]	[0.0048; 0.0102]	[0.0071; 0.0125]
70	[0.0027; 0.0081]	[0.0022; 0.0076]	[0.0002; 0.0056]	[0.0025; 0.0078]
80	[0.0007; 0.0061]	[0.0002; 0.0056]	[-0.0018; 0.0037]	[0.0005; 0.0059]
90	[-0.0001; 0.0054]	[-0.0005; 0.0049]	[-0.0026; 0.0029]	[-0.0003; 0.0051]
100	[-0.0009; 0.0044]	[-0.0015; 0.0039]	[-0.0035; 0.0019]	[-0.0012; 0.0042]
110	[-0.0013; 0.0041]	[-0.0018; 0.0036]	[-0.0038; 0.0016]	[-0.0015; 0.0038]
120	[-0.0002; 0.0052]	[-0.0007; 0.0047]	[-0.0028; 0.0027]	[-0.0005; 0.0050]
130	[0; 0]	[-0.0032; 0.0021]	[-0.0052; 0.0002]	[-0.0029; 0.0024]
140	[-0.0021; 0.0032]	[0; 0]	[-0.0047; 0.0007]	[-0.0024; 0.0029]
150	[-0.0002; 0.0052]	[-0.0007; 0.0047]	[0; 0]	[-0.0004; 0.0050]
160	[-0.0024; 0.0029]	[-0.0029; 0.0024]	[-0.0050; 0.0004]	[0; 0]
170	[-0.0022; 0.0033]	[-0.0027; 0.0027]	[-0.0047; 0.0008]	[-0.0024; 0.0030]
180	[-0.0010; 0.0045]	[-0.0015; 0.0039]	[-0.0035; 0.0020]	[-0.0012; 0.0042]

Tabla 6.3: Ajuste parámetro I para conjunto N°1 (Tabla #2).

Parámetro I	170	180
10	[0.9813; 1.0011]	[0.9802; 0.9999]
20	[0.2973; 0.3064]	[0.2961; 0.3052]
30	[0.0990; 0.1053]	[0.0978; 0.1041]
40	[0.0372; 0.0428]	[0.0360; 0.0416]
50	[0.0156; 0.0211]	[0.0144; 0.0199]
60	[0.0067; 0.0122]	[0.0055; 0.0111]
70	[0.0021; 0.0076]	[0.0010; 0.0064]
80	[0.0002; 0.0056]	[-0.0010; 0.0044]
90	[-0.0006; 0.0049]	[-0.0018; 0.0037]
100	[-0.0015; 0.0039]	[-0.0027; 0.0027]
110	[-0.0019; 0.0036]	[-0.0031; 0.0024]
120	[-0.0008; 0.0047]	[-0.0020; 0.0035]
130	[-0.0033; 0.0022]	[-0.0045; 0.0010]
140	[-0.0027; 0.0027]	[-0.0039; 0.0015]
150	[-0.0008; 0.0047]	[-0.0020; 0.0035]
160	[-0.0030; 0.0024]	[-0.0042; 0.0012]
170	[0; 0]	[-0.0039; 0.0015]
180	[-0.0015; 0.0039]	[0; 0]

Tabla 6.4: Ajuste parámetro I para conjunto N°1 (Tabla #3).

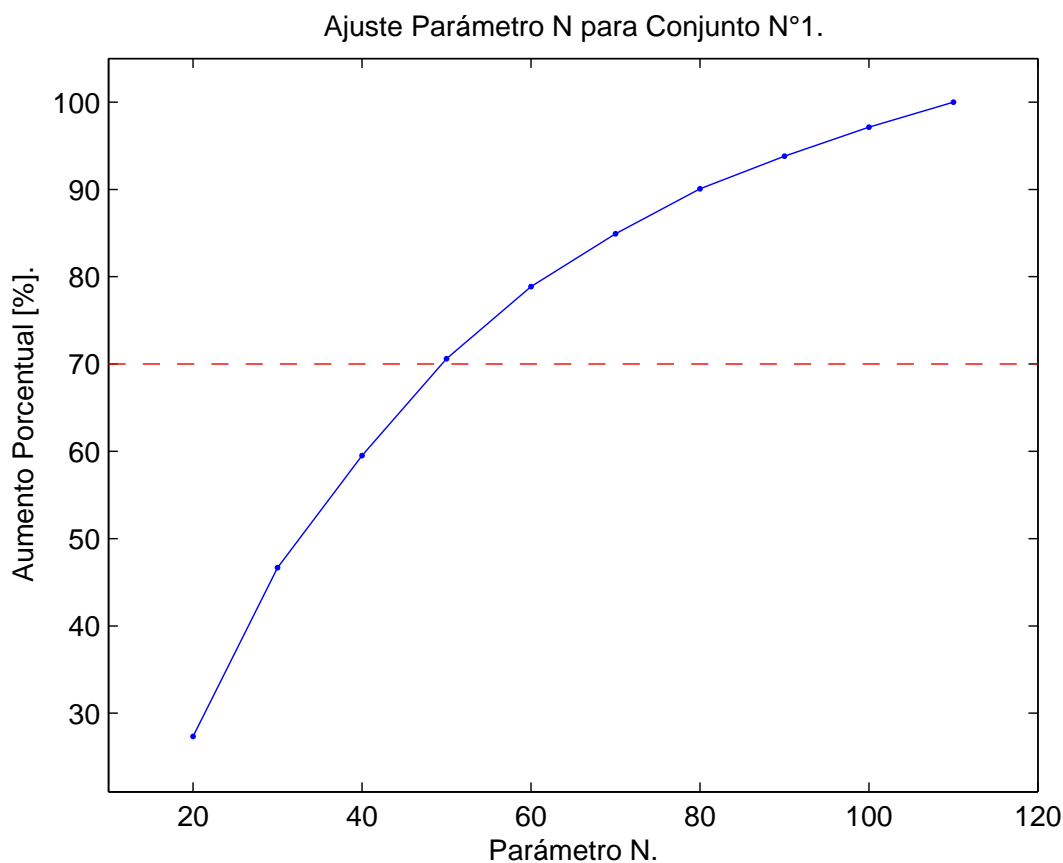


Figura 6.1: Ajuste parámetro N para conjunto N°1.

Parámetro N	10	20	30
10	[0; 0]	[0.0292; 0.0343]	[0.0518; 0.0566]
20	[-0.0343; -0.0292]	[0; 0]	[0.0204; 0.0245]
30	[-0.0566; -0.0518]	[-0.0245; -0.0204]	[0; 0]
40	[-0.0715; -0.0668]	[-0.0393; -0.0354]	[-0.0166; -0.0131]
50	[-0.0843; -0.0797]	[-0.0522; -0.0484]	[-0.0295; -0.0261]
60	[-0.0939; -0.0894]	[-0.0618; -0.0580]	[-0.0390; -0.0358]
70	[-0.1009; -0.0964]	[-0.0687; -0.0650]	[-0.0460; -0.0428]
80	[-0.1069; -0.1024]	[-0.0747; -0.0710]	[-0.0519; -0.0488]
90	[-0.1112; -0.1067]	[-0.0790; -0.0754]	[-0.0563; -0.0532]
100	[-0.1150; -0.1106]	[-0.0829; -0.0793]	[-0.0601; -0.0571]
110	[-0.1184; -0.1139]	[-0.0862; -0.0826]	[-0.0635; -0.0604]
	40	50	60
10	[0.0668; 0.0715]	[0.0797; 0.0843]	[0.0894; 0.0939]
20	[0.0354; 0.0393]	[0.0484; 0.0522]	[0.0580; 0.0618]
30	[0.0131; 0.0166]	[0.0261; 0.0295]	[0.0358; 0.0390]
40	[0; 0]	[0.0114; 0.0145]	[0.0210; 0.0240]
50	[-0.0145; -0.0114]	[0; 0]	[0.0083; 0.0109]
60	[-0.0240; -0.0210]	[-0.0109; -0.0083]	[0; 0]
70	[-0.0310; -0.0281]	[-0.0179; -0.0153]	[-0.0082; -0.0058]
80	[-0.0369; -0.0341]	[-0.0238; -0.0213]	[-0.0141; -0.0118]
90	[-0.0412; -0.0385]	[-0.0282; -0.0257]	[-0.0185; -0.0162]
100	[-0.0451; -0.0424]	[-0.0320; -0.0296]	[-0.0223; -0.0201]
110	[-0.0484; -0.0457]	[-0.0353; -0.0329]	[-0.0256; -0.0235]
	70	80	90
10	[0.0964; 0.1009]	[0.1024; 0.1069]	[0.1067; 0.1112]
20	[0.0650; 0.0687]	[0.0710; 0.0747]	[0.0754; 0.0790]
30	[0.0428; 0.0460]	[0.0488; 0.0519]	[0.0532; 0.0563]
40	[0.0281; 0.0310]	[0.0341; 0.0369]	[0.0385; 0.0412]
50	[0.0153; 0.0179]	[0.0213; 0.0238]	[0.0257; 0.0282]
60	[0.0058; 0.0082]	[0.0118; 0.0141]	[0.0162; 0.0185]
70	[0; 0]	[0.0049; 0.0070]	[0.0093; 0.0114]
80	[-0.0070; -0.0049]	[0; 0]	[0.0034; 0.0053]
90	[-0.0114; -0.0093]	[-0.0053; -0.0034]	[0; 0]
100	[-0.0152; -0.0132]	[-0.0092; -0.0073]	[-0.0047; -0.0030]
110	[-0.0185; -0.0165]	[-0.0125; -0.0106]	[-0.0080; -0.0063]

Tabla 6.5: Ajuste parámetro N para conjunto №1 (Tabla #1).

Parámetro N	100	110
10	[0.1106; 0.1150]	[0.1139; 0.1184]
20	[0.0793; 0.0829]	[0.0826; 0.0862]
30	[0.0571; 0.0601]	[0.0604; 0.0635]
40	[0.0424; 0.0451]	[0.0457; 0.0484]
50	[0.0296; 0.0320]	[0.0329; 0.0353]
60	[0.0201; 0.0223]	[0.0235; 0.0256]
70	[0.0132; 0.0152]	[0.0165; 0.0185]
80	[0.0073; 0.0092]	[0.0106; 0.0125]
90	[0.0030; 0.0047]	[0.0063; 0.0080]
100	[0; 0]	[0.0025; 0.0042]
110	[-0.0042; -0.0025]	[0; 0]

Tabla 6.6: Ajuste parámetro N para conjunto N°1 (Tabla #2).

Parámetro ρ	0.1	0.2	0.3
0.1	[0; 0]	[0.0285; 0.0333]	[0.0320; 0.0367]
0.2	[-0.0333; -0.0285]	[0; 0]	[0.0015; 0.0055]
0.3	[-0.0367; -0.0320]	[-0.0055; -0.0015]	[0; 0]
0.4	[-0.0281; -0.0233]	[0.0032; 0.0072]	[0.0068; 0.0105]
0.5	[-0.0117; -0.0068]	[0.0196; 0.0237]	[0.0232; 0.0271]
0.6	[0.0223; 0.0276]	[0.0535; 0.0582]	[0.0571; 0.0616]
0.7	[0.0885; 0.0947]	[0.1196; 0.1254]	[0.1231; 0.1288]
0.8	[0.2616; 0.2711]	[0.2925; 0.3020]	[0.2959; 0.3055]
0.9	[0.6235; 0.6397]	[0.6543; 0.6707]	[0.6577; 0.6742]
	0.4	0.5	0.6
0.1	[0.0233; 0.0281]	[0.0068; 0.0117]	[-0.0276; -0.0223]
0.2	[-0.0072; -0.0032]	[-0.0237; -0.0196]	[-0.0582; -0.0535]
0.3	[-0.0105; -0.0068]	[-0.0271; -0.0232]	[-0.0616; -0.0571]
0.4	[0; 0]	[-0.0184; -0.0146]	[-0.0529; -0.0485]
0.5	[0.0146; 0.0184]	[0; 0]	[-0.0365; -0.0320]
0.6	[0.0485; 0.0529]	[0.0320; 0.0365]	[0; 0]
0.7	[0.1145; 0.1201]	[0.0980; 0.1036]	[0.0637; 0.0695]
0.8	[0.2873; 0.2968]	[0.2708; 0.2804]	[0.2366; 0.2461]
0.9	[0.6491; 0.6656]	[0.6325; 0.6491]	[0.5984; 0.6148]
	0.7	0.8	0.9
0.1	[-0.0947; -0.0885]	[-0.2711; -0.2616]	[-0.6397; -0.6235]
0.2	[-0.1254; -0.1196]	[-0.3020; -0.2925]	[-0.6707; -0.6543]
0.3	[-0.1288; -0.1231]	[-0.3055; -0.2959]	[-0.6742; -0.6577]
0.4	[-0.1201; -0.1145]	[-0.2968; -0.2873]	[-0.6656; -0.6491]
0.5	[-0.1036; -0.0980]	[-0.2804; -0.2708]	[-0.6491; -0.6325]
0.6	[-0.0695; -0.0637]	[-0.2461; -0.2366]	[-0.6148; -0.5984]
0.7	[0; 0]	[-0.1794; -0.1701]	[-0.5479; -0.5321]
0.8	[0.1701; 0.1794]	[0; 0]	[-0.3722; -0.3583]
0.9	[0.5321; 0.5479]	[0.3583; 0.3722]	[0; 0]

Tabla 6.7: Ajuste parámetro ρ para conjunto N°1.