



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

EVALUANDO FORMULACIONES EQUIVALENTES DE JUEGOS DE SEGURIDAD
DE STACKELBERG

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES

KARLA ROSAS CUENTAS

PROFESOR GUÍA:
FERNANDO ORDOÑEZ PIZARRO

MIEMBROS DE LA COMISIÓN:
VÍCTOR BUCAREY LÓPEZ
RICHARD WEBER HAAS
OSCAR FIGUEROA ULLOA

SANTIAGO DE CHILE
ENERO 2016

RESUMEN DE LA TESIS
PARA OPTAR AL GRADO DE
MAGÍSTER EN GESTIÓN DE OPERACIONES
POR: KARLA ROSAS CUENTAS
FECHA: ENERO 2016
PROF. GUÍA: FERNANDO ORDOÑEZ

EVALUANDO FORMULACIONES EQUIVALENTES DE JUEGOS DE SEGURIDAD DE STACKELBERG

Las agencias de seguridad alrededor del mundo enfrentan el reto de proteger objetivos sensibles, tales como aeropuertos, puertos u otras infraestructuras críticas. Para ello, toman acciones preventivas y asignan recursos de defensa a patrullar un objetivo. El problema es que, por lo general, los recursos a su disposición son limitados y la cantidad de objetivos a defender puede llegar a ser muy grande, por lo que la labor de asignación de recursos puede llegar a ser muy compleja y costosa en términos computacionales.

Para enfrentar este problema, un enfoque es utilizar técnicas de teoría de juegos para captar el razonamiento entre el defensor y los atacantes o criminales, específicamente se usan los Juegos de Seguridad de Stackelberg, ó SSG por sus siglas en inglés. Los SSGs son parte esencial de varias aplicaciones actualmente en uso por diversas agencias de seguridad en Estados Unidos y su éxito ha generado que el interés por estos aumente, por lo que muchos autores han dado diversas contribuciones sobre este tema. Sin embargo, uno de los puntos que aún es un tema abierto de estudio es la escalabilidad de las formulaciones, ya que los modelos existentes tienen un bajo desempeño cuando el número de tipos de atacantes es grande, lo que resulta inadecuado para problemas complejos del mundo real.

Una instancia cualquiera de SSG tiene muchas posibles formulaciones, este trabajo de tesis estudia tres: (SSMG), (DOBSS-SSG) y (MIP-SSG). Estas formulaciones son equivalentes entre sí, que tienen diferencias en términos de tamaño y fortaleza, lo que hace que no sea evidente cuál es la de mejor desempeño, pero todas resuelven el problema, en ámbitos con múltiples tipos de atacantes, aprovechando la estructura propia de los juegos de seguridad donde la utilidad de los jugadores depende solamente de si el objetivo atacado esta protegido o no, logrando de esta manera una representación compacta y acelerando los tiempos de ejecución.

Este trabajo describe y analiza cada una de estas formulaciones comparándolas tanto desde el punto de vista de modelación como computacional. En el aspecto de modelación se muestra el efecto de incorporar una representación logarítmica de las las variables enteras y el efecto de restringir el espacio de soluciones. En el aspecto computacional, el enfoque esta en el rendimiento, en términos del tiempo de ejecución requerido para hallar una solución, sobre un conjunto amplio de instancias, poniendo énfasis en escalar tanto el número de tipos de atacantes como el número de objetivos a defender. Se compara las formulaciones entre sí y se detalla las ventajas de estas sobre las formulaciones presentes en la literatura.

A Matias y Alonzo.

Agradecimientos

Al profesor Fernando Ordoñez por su constante apoyo durante la realización de este trabajo de tesis. Muchas gracias por la oportunidad y por su buena disposición en todo momento.

A Alonzo, por su compañía y por todo el ánimo y el apoyo brindado durante todo el camino.

A Víctor Bucarey, Oscar Figueroa y Richard Weber por su buena disposición para formar parte de la comisión, especialmente a Víctor por ser el gestor de la oportunidad y por confiar en mi para el desarrollo de este trabajo.

Al Instituto de Sistemas Complejos de Ingeniería (ISCI) por financiar parcialmente este trabajo de tesis.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Propósito de la Tesis	4
1.3. Estructura de la Tesis	5
2. Revisión Bibliográfica	6
3. Los Juegos de Seguridad de Stackelberg (SSG)	9
3.1. SSGM	15
3.2. DOBSS-SSG	17
3.3. MIP-SSG	19
3.4. Formulaciones Logarítmicas	21
3.4.1. LSSGM	22
3.4.2. LDOBSS-SSG	22
3.4.3. LMIP-SSG	23
3.5. Incorporando restricciones al Espacio de Soluciones	23
3.5.1. Restricción a objetivos consecutivos	24
3.5.2. Restricción de objetivos en una misma estrategia	26
4. Estudio Computacional	29
4.1. Introducción	29
4.2. Tamaño de las diferentes formulaciones	31
4.3. Fortaleza de la relajación lineal	31
4.4. Rendimiento de las formulaciones	33
4.4.1. Tiempos de resolución	33
4.4.2. Efecto del orden de magnitud de las entradas de la matriz de utilidades	35
4.4.3. Efecto de la estructura de la matriz de utilidades	36
4.4.4. Efecto del número de tipos de atacantes	38
4.4.5. Efecto del número de objetivos	39
4.4.6. Efecto del número de recursos del defensor	41
4.5. Análisis de las formulaciones logarítmicas	43
4.6. Análisis de la restricción del espacio de soluciones del defensor	47

4.6.1. Efecto de restringir a locaciones consecutivas	47
4.6.2. Efecto de definir locaciones con prohibiciones	49
Conclusión	50
Bibliografía	55

Índice de Tablas

4.1. Información de las Instancias	30
4.2. Tamaño del set A cuando $ N = 40$, $m = (0,3 \cdot Q)$ y $ Q $ varía.	31
4.3. Fortaleza de la Relajación Lineal y Tiempo de ejecución del LP para el Set C	32
4.4. Tiempo de Ejecución del set A cuando $ L = 4$, $m = (0,3 \cdot Q)$ y $ Q $ varía	35
4.5. SSGM: variación del tamaño al incluir la representación logarítmica	44
4.6. DOBSS-SSG: variación del tamaño al incluir la representación logarítmica	44
4.7. MIP-SSG: variación del tamaño al incluir la representación logarítmica	45
4.8. Locaciones consecutivas vs caso irrestricto: tiempos de resolución.	49

Índice de Ilustraciones

4.1. Set A: Tiempo, Nodos y Gap vs % Acumulado de Problemas Resueltos. . . .	34
4.2. Efecto del orden de magnitud del 10 % de de datos.	36
4.3. Efecto de la estructura de la matriz de utilidades.	37
4.4. Efecto del número de tipos de atacantes $ L $	39
4.5. Efecto del número de atacantes $ L $ sin tiempo límite.	40
4.6. Efecto del número de objetivos $ Q $	41
4.7. Efecto del número de objetivos $ Q $ sin tiempo límite	42
4.8. Efecto del número de recursos del defensor $ m $	42
4.9. Efecto de la formulación logarítmica en el número de variables enteras. . . .	45
4.10. Perfil de las formulaciones SSGM versus LSSGM.	46
4.11. Perfil de las formulaciones DOBSS-SSG versus LDOBSS-SSG.	46
4.12. Perfil de las formulaciones MIP-SSG versus LMIP-SSG.	47
4.13. Tiempos de ejecución en el caso de restricción a objetivos consecutivos. . . .	48
4.14. Efecto de de definir objetivos con prohibiciones.	50

Capítulo 1

Introducción

1.1. Motivación

Las agencias de seguridad enfrentan el reto de proteger objetivos sensibles tales como aeropuertos, puertos u otras infraestructuras relevantes; sin embargo, los recursos a su disposición son limitados y esto impide proveer protección en todo momento [21]. Para enfrentar este problema, las agencias deben optar por un despliegue inteligente de recursos. Esto implica tomar en cuenta a posibles atacantes motivados que realicen algún tipo de seguimiento al defensor para planear ataques más efectivos [25].

Con el fin de modelar este tipo de problemas, una línea de investigación se basa en técnicas de Teoría de Juegos para captar el razonamiento entre agencias de seguridad y atacantes o criminales. Específicamente, se usan los Juegos de Seguridad de Stackelberg, que de aquí en adelante llamaremos SSG por sus siglas en inglés, donde un líder o defensor, se compromete a una estrategia que puede ser observada por el seguidor o atacante y luego este selecciona su estrategia. En este sentido, la asignación de recursos de seguridad es abstraído como un juego entre dos agentes, los cuales pueden no ser estrictamente personas, sino organizaciones y además estas pueden ser de más de un tipo.

Durante la última década, la investigación sobre los SSG ha dado lugar al desarrollo de distintos sistemas de apoyo a las agencias de seguridad para una efectiva asignación de recursos de seguridad, entre ellos esta ARMOR que actualmente esta en uso en el aeropuerto de Los Ángeles [34], GUARDS usado por la Administración de Seguridad del Transporte (TSA) para proteger a más de 400 aeropuertos en Estados Unidos [35], IRIS usado por el Servicio Federal de Alguaciles Aéreos para asignar personal de seguridad en vuelos internacionales [40], PROTECT para aleatorizar el patrullaje en los puertos y ferries de pasajeros en puertos tales como Nueva York, Boston y Los Ángeles [38], y TRUSTS usado por la policía de Los Ángeles para patrullar el metro [48]. Todas estas agencias de seguridad usan sistemas

computacionales basados en estos modelos de teoría de juegos para representar el problema de seguridad que enfrentan con el fin de cumplir los roles de prevención en los ámbitos que les corresponde. El uso de estas herramientas tiene evaluaciones positivas en estos distintos problemas de seguridad, ver [35], [40] y [38].

Para ilustrar el problema de seguridad consideremos el problema de patrullaje preventivo que enfrenta Carabineros de Chile como parte del Plan Cuadrante de Seguridad Preventiva (PCSP). El PCSP es la principal estrategia de operaciones urbanas de Carabineros, que está presente en 150 Comunas, con un total de 661 cuadrantes, la cual cubre un 85.4 % de la población nacional conforme al censo del Instituto Nacional de Estadísticas (INE) 2002. Esta estrategia al 2015 involucra a 434 Cuarteles, 151 Comisarías, 52 Subcomisarías, 70 Tenencias y 157 Retenes [11] Los servicios provistos por el PCSP se entienden como permanentes y efectivos en toda la jurisdicción de cualquier comisaría donde se implemente y según [44] estos son:

- Patrullajes preventivos focalizados, consisten en uno o más medios para ejercer vigilancia policial en terreno, con el propósito de evitar que se cometan ilícitos. La focalización consiste en que estos patrullajes son debidamente orientados tras un procesamiento de la información policial capturada en el cuadrante.
- Atención de procedimientos, es la actuación del personal en respuesta a una solicitud por quien lo requiera.
- Fiscalización de establecimientos, orientada a verificar que se cumplan las normativas legales vigentes, por ejemplo Ley de Control de Alcoholes.
- Cumplimiento de órdenes judiciales, es decir, de toda aquella orden de los tribunales, como por ejemplo arrestos o citaciones, entre otros.

Según [32], esta estrategia de Carabineros de Chile apuesta a la capacidad de gestionar el despliegue de servicios en un territorio sobre la base de la determinación de la oferta policial y la demanda de seguridad como la determinación de niveles de vigilancia en cada cuadrante, basada en la inteligencia producto de la información histórica captada por Carabineros en conjunto con el conocimiento de los agentes expertos de la institución y del uso de tecnología como por ejemplo el sistema de geolocalización de los delitos. Los recursos policiales aseguran estos cuatro servicios pero no son los suficientes para brindar protección preventiva permanente en todo lugar, a toda hora, en todos los días del año. Consecuentemente, Carabineros debe optar por asignar sus recursos de la mejor forma posible y esta labor puede llegar a ser de gran complejidad.

Tomando el estudio realizado en [22], el autor plantea un problema acotado a la 1era Comisaría de Santiago, basándose en un subconjunto de 6725 denuncias registradas en AUPOL comprendidas entre los años 2002 y 2004, con estos datos determina 119 objetivos o hot spots y estima que para el caso del servicio de patrullaje preventivo se cuenta con una dotación de pares de carabineros a pie de 23, 24 y 9 unidades en cada uno de los tres turnos de la comisaría, respectivamente. Este trabajo es un problema muy acotado a lo que

en realidad enfrenta Carabineros; sin embargo, incluso en este escenario menor, la cantidad de estrategias puras de defensa posibles para el turno 2 es de $\binom{119}{24} = \frac{119!}{24! \cdot 95!} = 8,6 \cdot 10^{24}$, definiendo una estrategia pura de defensa como la asignación de los carabineros disponibles a los lugares a proteger durante segmentos de tiempo dentro de cada turno; es decir, incluso en un escenario acotado de un caso real la cantidad de posibles estrategias de defensa es muy grande y complejo de decidir.

La asignación de recursos de seguridad en las aplicaciones de seguridad mencionadas, en el escenario mencionado de Carabineros de Chile y en otros escenarios de la vida real, involucran evaluar todas las posibles combinaciones de asignación de recursos a objetivos, y este número de combinaciones es por lo general grande y crece exponencialmente al aumentar el número de objetivos. Por lo tanto, generar una asignación de recursos efectiva es una tarea computacional costosa.

Un enfoque de resolución para este problema es aplicar técnicas de branch and price [17] y generación de columnas [2] para evitar la enumeración explícita del gran número de estrategias puras del defensor. Otro enfoque es aprovechar la estructura del problema mediante formulaciones exactas de programación entera mixta, MILP por sus siglas en inglés, que aprovechan la particularidad de los juegos de seguridad donde la utilidad de los jugadores depende solamente de si el objetivo es atacado o no y si ese objetivo esta protegido o no, y así se puede lograr una representación compacta que resuelve dando un vector óptimo de protección para cada objetivo o locación, reduciendo de esta manera el espacio de solución del defensor y acelerando los tiempos de resolución.

Las formulaciones de interés para este trabajo de tesis usan este último enfoque y consideran ámbitos donde el defensor se tiene que enfrentar a más de un tipo de atacante, que es por lo general algo natural en los temas de seguridad. Volviendo al ejemplo de Carabineros, al proteger un cuadrante ellos se pueden enfrentar a delincuentes oportunistas, traficantes de droga, allanadores de propiedades privadas e incluso terroristas dispuestos a colocar bombas para sembrar el pánico. Todos estos grupos o tipos de atacantes tienen distintas capacidades y el defensor no sabe a cuál de ellos se enfrentará en un determinado momento, pero dados los registros históricos de los delitos sí sabe la distribución de probabilidad de ocurrencia de los tipos de atacantes. Cuando se incluye esta incertidumbre sobre los tipos de atacantes, los juegos son denominados Bayesianos.

Una instancia cualquiera de este tipo de SSG tiene muchas formulaciones posibles. Es importante estudiar las formulaciones para saber cuál funciona mejor porque solo así se puede identificar aquellas propiedades invariantes a diferentes ámbitos pero que impactan en el desempeño, es decir, en el tiempo de ejecución requerido para calcular soluciones a una instancia del problema. Un entendimiento del tiempo requerido por las formulaciones permite profundizar su uso en problemas a gran escala, es decir, en aplicaciones del mundo real.

En este trabajo se estudian tres formulaciones equivalentes, las dos primeras formulaciones, denominadas (SSGM) y (DOBSS-SSG), usan la técnica de la gran M y tienen menos variables que la tercera formulación, denominada (MIP-SSG), que no usa esta técnica, ya que por lo general suele dar malas relajaciones lineales, pero en su lugar usa más restricciones. Es este balance entre tamaño y fortaleza de las formulaciones que hace que no sea evidente cuál formulación tendrá un mejor desempeño en términos de tiempo de ejecución. A lo largo de este trabajo se enfatiza en las potenciales ventajas de cada una de las formulaciones. Es importante notar que dada la complejidad de los solvers de optimización del estado del arte, es difícil predecir con alta precisión cuál formulación es mejor. Algunas pautas pueden ser encontradas en estudios computacionales, como por ejemplo [43], pero la formulación que se desempeña mejor puede depender fuertemente de la estructura específica del problema o de los datos. Afortunadamente, hay una alta correlación entre ciertas propiedades de una formulación y un buen rendimiento computacional.

1.2. Propósito de la Tesis

En esta tesis se revisa y compara tres formulaciones de programación entera mixta para los SSG, denominadas (SSGM), (DOBSS-SSG) y (MIP-SSG). Se considera tanto aspectos de modelación como computacionales. En términos de modelamiento se discute formulaciones que utilizan un número logarítmico de variables enteras y la restricción del espacio de estrategias factibles del defensor. En términos de aspectos computacionales el enfoque está en el rendimiento, en términos de tiempos de ejecución.

Se sabe que la mayoría de problemas de programación entera mixta pueden ser formulados de diferentes maneras y que las propiedades de estas diferentes formulaciones pueden afectar significativamente su rendimiento. Uno de estos aspectos es el tamaño de la formulación en términos de número de variables y restricciones. Otro es el aspecto del ajuste de la formulación. Hay muchas formas de definir la noción de ajuste. Dado que en esta tesis se consideran formulaciones que están definidas con diferentes variables, se compara también sus valores de la relajación lineal. Es decir, el valor óptimo que se obtiene cuando se resuelve una formulación sin imponer las restricciones de integralidad. Esto define una cota natural en el valor de la optimización, e indica que tan buen referente es el valor de la relajación lineal con respecto al valor óptimo real. Esto es útil ya que el gap de integralidad es un buen predictor de la dificultad computacional de un algoritmo de Branch and Bound. Para mayor detalle ver [29]. Una medida final que se considera es el tiempo que toma encontrar una solución óptima, la sensibilidad ante el cambio de los parámetros y, cuando se impone un lapso de tiempo razonable, se considera la proporción del total de instancias que encuentran una solución óptima dentro del límite establecido.

1.3. Estructura de la Tesis

La tesis esta organizada de la siguiente manera: En el Capítulo 2 se da una breve descripción sobre el trabajo relacionado y una revisión del rendimiento que obtuvieron los modelos y algoritmos relacionados. En el Capítulo 3 se da una introducción sobre los Juegos de Seguridad de Stackelberg (SSG) y la notación que va a ser utilizada. Luego, se describe las tres formulaciones de programación mixta entera para SSG y, se discute la incorporación de extensiones a los modelos en cuanto a la reducción del número de variables enteras usando formulaciones logarítmicas y la restricción del espacio de soluciones factibles del defensor. En el Capítulo 4 se presenta los resultados computacionales obtenidos al evaluar estos enfoques usando las instancias generadas aleatoriamente. El análisis se enfoca en el tamaño de las diferentes formulaciones, la fortaleza de la cota que da la relajación lineal, la dificultad de revolver cada una de las formulaciones a optimalidad, la sensibilidad de los modelos ante cambios en sus parámetros, el efecto de incorporar la representación logarítmica y la restricción del espacio de soluciones. Finalmente, en el Capítulo 5 se discute las ventajas relativas de cada formulación presentadas como conclusiones finales.

Capítulo 2

Revisión Bibliográfica

La idea principal de este capítulo es discutir el estado del arte de los Juegos de Seguridad, describir brevemente las aplicaciones que están en uso y presentar el rendimiento de los modelos y algoritmos desarrollados.

El Departamento de Seguridad Interna de Estados Unidos ha implementado exitosamente diversas aplicaciones basadas en juegos de Stackelberg, por ejemplo [20], [38], [48], entre otras, lo que ha generado que el interés hacia estos juegos aumente. Muchos autores han dado contribuciones prácticas y teóricas al campo de Teoría de Juegos de Stackelberg.

Entre las contribuciones prácticas tenemos el desarrollo de aplicaciones categorizadas en:

- Juegos de seguridad en infraestructura, el objetivo es asistir a las autoridades en actividades de contra-terrorismo, ver por ejemplo [40] y [34].
- Juegos de seguridad en temas ambientales, aquí se busca proteger recursos naturales contra agresores ilegales que tienen movilidad dentro de un área geográfica, como el caso de cazadores furtivos descrito en [45] y el caso de la pesca ilegal descrito en [15].
- Juegos de seguridad en ámbitos fuera de la seguridad, por ejemplo en [27] donde los juegos de Stackelberg se usan en pruebas de software para detectar programas maliciosos y, en [6] y [5] donde se usa para auditar el acceso a información sensible como la que manejan los hospitales.

Muchos autores han trabajado en clasificar los juegos generales de Stackelberg así como los SSG en términos de su complejidad computacional [26], [10].

Los juegos generales han sido clasificados completamente. Los juegos generales con un solo tipo de atacante pueden ser resueltos en tiempo polinomial aplicando una técnica conocida como múltiples LPs, enfoque diseñado por [10]. Asimismo, los autores de [9] entregaron un programa lineal (LP) compacto con un número polinomial de variables y restricciones,

mejorando el MILP propuesto en [10]. Su LP es una descripción lineal de la envoltura convexa de las soluciones factibles

Cuando se extiende estos juegos de un tipo de atacante a varios tipos de atacantes, en [10] se prueba que los juegos se convierten en NP-hard y se provee una reducción a un problema NP-hard conocido como el problema del conjunto independiente.

Representar el ámbito de los problemas de seguridad como Juegos de Stackelberg llevó a definir los SSGs. En estos juegos se tiene que lidiar con la complicación adicional de tener un defensor con un número exponencial de estrategias puras de asignación de recursos de seguridad para proteger objetivos. Cuando se trata con SSGs en los cuales hay un solo tipo de atacantes, en [26] se afirma que la complejidad de los juegos depende de si los recursos de seguridad son homogéneos, aquellos que pueden proteger cualquier objetivo, o heterogéneos, aquellos que solo pueden proteger ciertos tipos de objetivos, y en el número de objetivos que un recurso puede proteger simultáneamente.

Cuando los recursos pueden proteger solo objetivos individuales, independientemente que estos sean homogéneos o heterogéneos, los juegos pueden resolverse en tiempo polinomial. Si los recursos son homogéneos y pueden proteger dos objetivos simultáneamente, en [26] también se muestra que estos SSGs pueden ser polinomiales. En cualquier otro caso ellos probaron que los otros SSGs son NP-hard.

Cuando se extiende estos SSGs al caso de varios tipos de atacantes, pues cualquier caso que era NP-hard con un solo tipo de atacantes, se mantiene NP-hard. Sin embargo, los casos polinomiales no necesariamente se mantienen polinomiales.

Muchos autores también han contribuido incluyendo la modelación del comportamiento humano en sus enfoques. Por ejemplo, en [47] y [8] se incluye en los modelos la racionalidad limitada de los atacantes.

Otros han contribuido con métodos de solución que se enfocan en la resolución de juegos de Stackelberg que son NP-hard y que varían dependiendo del contexto. Cuando se trata del caso general un enfoque de solución involucra la transformación de Harsanyi [14]. Esta reduce el caso de múltiples tipos de atacantes a un caso con un solo tipo de atacante, con un número exponencial de estrategias puras. Esto luego puede ser resuelto usando el método de múltiples LP propuesto por [10] que, en este caso, requiere resolver un número exponencial de LPs.

En [30] se introduce (DOBSS) aprovechando la propiedad de que los tipos de atacantes son independientes. La clave en (DOBSS) está en que evaluar la estrategia del defensor contra la matriz del juego usando la transformación de Harsanyi es equivalente a evaluar la estrategia del defensor contra cada una de las matrices del juego para los tipos de atacantes de manera individual y luego obtener una suma ponderada y así evita resolver un número exponencial de LPs del juego que usa la transformación de Harsanyi. Sin embargo, si el defensor tiene

un número exponencial de estrategias puras, (DOBSS) es computacionalmente ineficiente. Ante este problema de escalabilidad, en [25] proponen (ERASER) que permite resolver en tiempo corto instancias grandes, esta formulación produce un vector de probabilidades de cobertura óptima para los objetivos. Estas probabilidades son luego usadas para llegar en tiempo polinomial a la estrategia mixta óptima del defensor.

Si bien (ERASER) logra escalar a problemas grandes, este tiene limitaciones cuando no todas las combinaciones de asignación son permitidas, por ejemplo en el caso del Servicio Federal de Alguaciles Aéreos (FAMS), la asignación de recursos debe tomar en cuenta consideraciones espacio-temporales para asignar alguaciles a vuelos, es decir, solo se puede asignar un alguacil a un vuelo si este está previamente en el sitio de despegue a la hora del vuelo, ante este tipo de restricciones, en [25] proponen (ERASER-C) que resuelve el problema con estas restricciones pero está limitado a pares de tramos, por ejemplo Nueva York-Londres, Londres-Nueva York, y no da soluciones óptimas para tramos de vuelos más complejos. Ante esto en [19] proponen (ASPEN) que resuelve para las estrategias puras de defensa y no para las probabilidades marginales de protección como (ERASER) y evita la enumeración explícita de las mismas usando técnicas de generación de columnas. Otras formas de evitar la enumeración explícita de estrategias puras es aplicando técnicas de Branch and Price [2] como el utilizado en [17] y, también se aplica técnicas de planos de corte [24] como el utilizado en [46].

Finalmente, dejando el ámbito de modelación estático, también hay trabajo relacionado al desarrollo de técnicas que involucran modelos dinámicos de optimización, las cuales consideran el hecho que el atacante puede ajustar su comportamiento de acuerdo a si recibe información nueva que le sugiera una mejor estrategia. Aquí tenemos como ejemplos [37] y [31], ambos se basan en aprendizaje y usan Procesos de Decisión de Markov (MPD por sus siglas en inglés) y Procesos de Decisión de Markov Parcialmente Observables (POMDP por sus siglas en inglés). Como parte del trabajo en [37] se modela el problema de patrullaje de locaciones, variando los ratios de incidencia de delitos en cada locación para luego obtener una ruta óptima usando el enfoque MDP.

Capítulo 3

Los Juegos de Seguridad de Stackelberg (SSG)

Los Juegos de Seguridad de Stackelberg o SSG son juegos entre un defensor y un atacante que puede ser de distintos tipos donde el defensor primero se compromete con una estrategia mixta que toma en cuenta la mejor respuesta del atacante, dado que este lo observa [10], [30], [25]. Las acciones del defensor corresponden a las diferentes medidas de seguridad que ella puede adoptar, cada acción del defensor es considerada una *estrategia pura*, aleatorizar sobre las estrategias puras da una *estrategia mixta*. El espacio de acción del atacante es la elección del objetivo a atacar, esta elección es una estrategia pura.

El defensor siempre puede aspirar a una mayor utilidad jugando estrategias mixtas, mientras que el atacante siempre tendrá una estrategia pura que es óptima [30].

Cada objetivo esta asociado a cuatro valores de utilidad: ganancia y pérdida para el defensor y el atacante respectivamente. Para el defensor es ganancia si el objetivo atacado esta protegido y pérdida si el objetivo atacado no esta protegido; para el atacante es de manera inversa [18]. Esta estructura de las utilidades ha sido utilizada ampliamente en trabajos previos y es clave, por ejemplo, para poder resolver problemas de gran tamaño como el del Servicio Federal de Alguaciles Aéreos [40].

Un juego de seguridad de Stackelberg bayesiano extiende el marco a múltiples tipos de atacantes, cada uno de estos tipos identificados por sus propias matrices de utilidades y cada cual con los cuatro valores por objetivo, como se explica anteriormente. El defensor no sabe a cual atacante enfrentará en una instancia dada, pero sabe la distribución de probabilidad de la cual el tipo de atacante es elegido [25].

A lo largo de esta tesis se considera la siguiente notación para describir el problema formalmente. Sea L el conjunto de tipos de atacantes, Q el conjunto de objetivos a defender

y m el número de recursos de seguridad del defensor. La probabilidad de los tipos de atacantes se representa por p^l y corresponde a una distribución dada a priori. Las utilidades del defensor y atacante se denotan de la siguiente manera:

- Rd_j^l Recompensa del Defensor dado que la estrategia de defensa protege el objetivo j y el atacante es del tipo l .
- Pd_j^l Pérdida del Defensor dado que la estrategia de defensa no protege el objetivo j y el atacante es del tipo l .
- Ra_j^l Recompensa del Atacante dado que la estrategia de defensa no protege el objetivo j y el atacante es del tipo l .
- Pa_j^l Pérdida del Atacante dado que la estrategia de defensa protege el objetivo j y el atacante es del tipo l .

Considerar que $Rd_j^l > Pd_j^l, \forall l \in L, \forall j \in Q$ y lo mismo para el atacante $Ra_j^l > Pa_j^l, \forall l \in L, \forall j \in Q$, es decir, colocar un recurso de seguridad para proteger un objetivo ayuda al defensor y perjudica al atacante; mientras que dejar un objetivo desprotegido ayuda al atacante.

El juego no es de suma cero, y es así también en aplicaciones en uso como la del aeropuerto de los Ángeles [33], ya que la utilidad de los atacantes puede depender de factores que podrían no tener relevancia en la utilidad del defensor, factores como el costo de montar el ataque, el valor de la propaganda de un ataque, entre otros. Por ejemplo, incluso si un atacante es arrestado, este podría considerar la publicidad de su detención como un atenuante. Es importante notar además que en [4], [23] y [36] explícitamente destacan que los intereses del defensor y los atacantes no son exactamente opuestos uno de otro y que los intereses de los diferentes tipos o grupos de atacantes son por lo general distintos entre sí. Consecuentemente este trabajo considera diferencias entre las utilidades del defensor y de los atacantes y entre los tipos de atacantes.

Los SSG son un tipo particular de juegos de Stackelberg adaptados al ámbito de seguridad donde una estrategia del defensor consiste en proteger un subconjunto de objetivos del tamaño de número de recursos que tiene a su disposición. Como ya se mencionó las utilidades de los jugadores en estos juegos depende solamente de si el objetivo es atacado o no y si este está protegido o no. Esta representación de las matrices de utilidades es una forma compacta de representación de la matriz de los juegos de Stackelberg generales.

Una dificultad que surge en este contexto es que las estrategias puras del defensor son exponenciales en número ya que corresponden a todas las posibles combinaciones de asignación de recursos a objetivos, esto significa que las formulaciones para los SSG pueden involucrar un número exponencial de variables y restricciones.

Las formulaciones estudiadas en esta tesis aprovechan dicha particularidad de las matrices para usar un número polinomial de variables que representen las estrategias puras del defen-

sor, evitando así la carga computacional de manejar un número exponencial de estrategias puras. Una vez obtenida la solución de estas formulaciones, uno puede recuperar las variables originales del defensor usadas para construir las estrategias mixtas en tiempo polinomial ya que esto constituye resolver un sistema lineal factible con un número de desigualdades igual al número de objetivos más uno.

A modo de tener claridad sobre las formulaciones, a continuación se presenta brevemente como estos MILP fueron obtenidos a partir del caso general de los juegos de Stackelberg. Las dos primeras, (SSGM) y (DOBSS-SSG) son el resultado de reformular (ERASER) y (DOBSS), dos modelos desarrollados previamente en la literatura, ver [30] y [25] respectivamente; y la tercera formulación, (MIP-SSG), es una formulación equivalente de (DOBSS-SSG) que no usa las restricciones del tipo M grande. Luego, se describe las extensiones a estos modelos usando una representación logarítmica de los objetivos y limitando el espacio de soluciones factibles para el defensor.

Consideremos primero el problema del atacante de forma general. El atacante observa la estrategia del defensor y acordemente selecciona un objetivo a atacar que maximice su utilidad. Sea X el conjunto de estrategias puras del defensor, R la matriz de utilidades del defensor y C la matriz de utilidades del atacante, en estas matrices la entrada i, j corresponde a que el defensor seleccione la estrategia pura i y el atacante elija la j . Si fijamos la estrategia del defensor a una $x \in X$, el problema del atacante puede ser representado de la siguiente manera:

$$\begin{aligned}
 \text{Max}_q \quad & \sum_{i \in X} \sum_{j \in Q} C_{ij} \cdot x_i \cdot q_j \\
 \text{s.t.} \quad & \sum_{j \in Q} q_j = 1 \\
 & q_j \geq 0 \quad \forall j \in Q
 \end{aligned} \tag{3.1}$$

Como este problema considera un problema lineal, o LP por sus siglas en inglés, cuyo conjunto factible forma un poliedro acotado donde todos los puntos extremos son estrategias puras, existe un óptimo que es un punto extremo [3]; es decir, siempre existe una estrategia pura que es óptima. Por ello, se considera que siempre el atacante maximiza su utilidad con una estrategia pura.

Por otro lado, el problema del defensor es elegir una estrategia que maximice su utilidad dado que el atacante nos observa. Este problema puede ser expresado como un problema de optimización bi-nivel, o (BP) por sus siglas en inglés, introducido en [7], el cual captura la secuencialidad y jerarquía de los juegos de Stackelberg. El BP apunta a problemas de optimización en los cuales parte de las restricciones representa que algunas de las variables constituyen una solución óptima a un problema de optimización anidado. En los juegos de

Stackelberg el primer nivel se refiere al problema del líder y el segundo nivel se refiere al problema del seguidor. El (BP) se puede representar de la siguiente manera:

$$\begin{aligned}
Max_{x,q} \quad & \sum_{i \in X} \sum_{j \in Q} R_{ij} \cdot x_i \cdot q_j \\
s.t. \quad & \sum_{i \in X} x_i = 1 \\
& x \geq 0 \\
q = \operatorname{argmin}_g \quad & \sum_{i \in X} \sum_{j \in Q} C_{ij} \cdot x_i \cdot g_j : \quad \sum_{j \in Q} g_j = 1, \quad g \geq 0
\end{aligned} \tag{3.2}$$

Este (BP) puede ser reducido a una formulación entera mixta cuadrática, o (QUAD) por sus siglas en inglés, de un solo nivel. Las condiciones de optimalidad del LP (3.1) caracterizan la solución óptima del problema del seguidor y estas son: restricciones de factibilidad primal, restricciones de factibilidad dual y holgura complementaria. Estas condiciones pueden ser usadas en el problema de optimización del líder para caracterizar la mejor respuesta del seguidor, resultando en la siguiente formulación:

$$\begin{aligned}
Max_{x,q,a} \quad & \sum_{i \in X} \sum_{j \in Q} R_{ij} \cdot x_i \cdot q_j \\
s.t. \quad & \sum_{i \in X} x_i = 1 \\
& x \geq 0 \\
& 0 \leq a - \sum_{i \in X} C_{ij} \cdot x_i \leq M \cdot (1 - q_j) \quad \forall j \in Q \\
& \sum_{j \in Q} q_j = 1 \\
& q \in \{0, 1\}^{|Q|}
\end{aligned} \tag{3.3}$$

Dado que en este trabajo se considera que el defensor se enfrenta a un conjunto de L tipos de atacantes, con una probabilidad de ocurrencia conocida a priori $p^l \geq 0 \quad \forall l \in L$, con $\sum_{l \in L} p^l = 1$, las utilidades dependen de cuál es el atacante que enfrenta el defensor y el objetivo es optimizar la utilidad esperada dada la distribución de probabilidades de los tipos de atacantes, este tipo de juegos es denominado Bayesiano, modificando la formulación anterior obtenemos la siguiente formulación cuadrática o (QUAD):

$$\begin{aligned}
Max_{x,q,a} \quad & \sum_{l \in L} \sum_{i \in X} \sum_{j \in Q} R_{ij}^l \cdot x_i \cdot q_j^l \\
s.t. \quad & \sum_{i \in X} x_i = 1 \\
& x \geq 0 \\
& 0 \leq a^l - \sum_{i \in X} C_{ij}^l \cdot x_i \leq M \cdot (1 - q_j^l) \quad \forall j \in Q, \forall l \in L \\
& \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \\
& q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L
\end{aligned} \tag{3.4}$$

Para enfrentar la carga de la no-linearidad de la función objetivo en (QUAD), la literatura usa dos enfoques.

El primer enfoque introduce variables reales para representar una cota inferior de la utilidad del líder y reformula la función objetivo en tal forma que conlleva a la formulación MILP denominada (ERASER), ver [25] para más detalles. Para el caso de múltiples atacantes (ERASER) queda así:

$$Max_{x,q,a,d} \quad \sum_{l \in L} p^l \cdot d^l$$

$$s.t. \quad d^l \leq \sum_{i \in X} R_{ij}^l \cdot x_i + M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \tag{3.5a}$$

$$\sum_{i \in X} x_i = 1 \tag{3.5b}$$

$$0 \leq a^l - \sum_{i \in X} C_{ij}^l \cdot x_i \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \tag{3.5c}$$

$$\sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \tag{3.5d}$$

$$q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \tag{3.5e}$$

El segundo enfoque, lineariza el término cuadrático no convexo de la función objetivo a través del uso de variables que representen el producto de las variables originales. Esto conlleva a (DOBSS), ver [30] para más detalles. Como se explica en ese trabajo, esta linealización es válida porque una solución factible de (QUAD) conlleva a una solución factible de (DOBSS) con el mismo valor objetivo y viceversa. Sea $z_{ij}^l = x_i \cdot q_j^l \quad \forall i \in X, \forall j \in Q, \forall l \in L$, (DOBSS) queda así:

$$\begin{aligned}
Max_{q,z,a} \quad & \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l \cdot R_{ij}^l \cdot z_{ij}^l \\
s.t. \quad & \sum_{i \in X} \sum_{j \in Q} z_{ij}^l = 1 & \forall l \in L & (3.6a) \\
& \sum_{j \in Q} z_{ij}^l \leq 1 & \forall l \in L, \forall i \in X & (3.6b) \\
& \sum_{i \in X} z_{ij}^l \geq q_j^l & \forall l \in L, \forall j \in Q & (3.6c) \\
& \sum_{i \in X} z_{ij}^l \leq 1 & \forall l \in L, \forall j \in Q & (3.6d) \\
& \sum_{j \in Q} q_j^l = 1 & \forall l \in L & (3.6e) \\
& q^l \in \{0, 1\}^{|Q|} & \forall l \in L & (3.6f) \\
& 0 \leq a^l - \sum_{i \in X} \sum_{k \in Q} C_{ij}^l \cdot z_{ik}^l \leq (1 - q_j^l) \cdot M & \forall l \in L, \forall j, h \in Q & (3.6g) \\
& \sum_{j \in Q} z_{ij}^l = \sum_{j \in Q} z_{ij}^1 & \forall l \in L, \forall i \in X & (3.6h) \\
& z_{ij}^l \geq 0 & \forall l \in L, \forall i \in X, \forall j \in Q & (3.6i) \\
& a^l \in \Re & & (3.6j)
\end{aligned}$$

Notar que la restricción (3.6c) puede ser reemplazadas por $\sum_{i \in X} z_{ij}^l = q_j^l$ considerando la restricción (3.6e) y que la restricción (3.6b) esta implicada por la no-negatividad de z y

por la restricción (3.6a). Haciendo estos cambios este MILP queda de la siguiente manera:

$$Max_{q,z,a} \quad \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l \cdot R_{ij}^l \cdot z_{ij}^l$$

$$s.t. \quad \sum_{i \in X} \sum_{j \in Q} z_{ij}^l = 1 \quad \forall l \in L \quad (3.7a)$$

$$\sum_{i \in X} z_{ij}^l = q_j^l \quad \forall l \in L, \forall j \in Q \quad (3.7b)$$

$$\sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.7c)$$

$$\sum_{j \in Q} z_{ij}^l = \sum_{j \in Q} z_{ij}^1 \quad \forall l \in L, \forall i \in X \quad (3.7d)$$

$$z_{ij}^l \geq 0 \quad \forall l \in L, \forall i \in X, \forall j \in Q \quad (3.7e)$$

$$a^l \in \mathfrak{R} \quad (3.7f)$$

$$q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.7g)$$

$$0 \leq a^l - \sum_{i \in X} \sum_{k \in Q} C_{ij}^l \cdot z_{ik}^l \leq (1 - q_j^l) \cdot M \quad \forall l \in L, \forall j, h \in Q \quad (3.7h)$$

Ahora aprovechemos la particularidad de las matrices de utilidades del caso de seguridad.

Notemos que:

$$R_{ij}^l = \begin{cases} Rd_j^l & \text{si } j \in i, \\ Pd_j^l & \text{si } j \notin i \end{cases}, \quad C_{ij}^l = \begin{cases} Pa_j^l & \text{si } j \in i, \\ Ra_j^l & \text{si } j \notin i \end{cases} \quad (3.8)$$

Si consideramos un juego donde el defensor debe decidir a cuales de los N objetivos asignar los m recursos de seguridad disponibles ($m \leq N$) y que el atacante elige atacar uno de los N objetivos, entonces el número de estrategias del atacante es $|Q| = N$ y el número de estrategias del defensor es $|X| = \binom{N}{m}$. Se asume que un recurso de seguridad puede ser asignado a lo más a un objetivo.

Con esto podemos reformular los MILP (3.5) y (3.7) para obtener las formulaciones de interés para esta tesis:

3.1. SSGM

Si notamos que los términos $\sum_{i \in X} R_{ij}^l \cdot x_i$ y $\sum_{i \in X} C_{ij}^l \cdot x_i$ pueden ser reescritos usando la equivalencia mostrada en (3.8) y si además definimos un vector de protección y igual a la

frecuencia total de protección del objetivo j de la siguiente manera:

$$y_j = \sum_{i \in X: j \in i} x_i \quad (3.9)$$

Podemos reescribir las restricciones (3.5a) y (3.5c) como se muestra en (3.10) y (3.11) respectivamente.

$$\begin{aligned} \sum_{i \in X} R_{ij}^l \cdot x_i &= Rd_j^l \cdot \sum_{i \in X: j \in i} x_i + Pd_j^l \cdot \sum_{i \in X: j \notin i} x_i \\ &= Rd_j^l \cdot y_j + Pd_j^l \cdot (1 - y_j) \end{aligned} \quad (3.10)$$

$$\begin{aligned} \sum_{i \in X} C_{ij}^l \cdot x_i &= Pa_j^l \cdot \sum_{i \in X: j \in i} x_i + Ra_j^l \cdot \sum_{i \in X: j \notin i} x_i \\ &= Pa_j^l \cdot y_j + Ra_j^l \cdot (1 - y_j) \end{aligned} \quad (3.11)$$

Usando (3.9), (3.10) y (3.11), el MILP que se obtiene es (3.12).

$$\begin{aligned} Max_{q,a,d} \quad & \sum_{l \in L} p^l \cdot d^l \\ s.t. \quad & d^l \leq Rd_j^l \cdot y_j + Pd_j^l \cdot (1 - y_j) + M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.12a) \\ & 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.12b) \\ & \sum_{j \in Q} y_j = m \quad (3.12c) \\ & 0 \leq y \leq 1 \quad (3.12d) \\ & \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.12e) \\ & q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.12f) \\ & \sum_{i \in X} x_i = 1 \quad (3.12g) \\ & x \geq 0 \quad (3.12h) \\ & y_j = \sum_{i \in X: j \in i} x_i \quad \forall j \in Q \quad (3.12i) \end{aligned}$$

Si omitimos las restricciones (3.12g), (3.12h) y (3.12i) para evitar considerar el gran número de estrategias puras del defensor, correspondiente al conjunto X , obtenemos el primer modelo denominado (SSGM), ver formulación (3.13).

$$\begin{aligned}
Max_{q,a,d} \quad & \sum_{l \in L} p^l \cdot d^l \\
s.t. \quad & d^l \leq Rd_j^l \cdot y_j + Pd_j^l \cdot (1 - y_j) + M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.13a) \\
& 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.13b) \\
& \sum_{j \in Q} y_j = m \quad (3.13c) \\
& 0 \leq y \leq 1 \quad (3.13d) \\
& \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.13e) \\
& q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.13f)
\end{aligned}$$

Notar que (SSGM) resuelve para un vector óptimo de probabilidades de protección de un objetivo y el vector óptimo de ataque para el atacante. Recuperar la estrategia mixta óptima para el defensor en base a las estrategias puras originales puede ser hecho en tiempo polinomial ya que es equivalente a resolver un sistema lineal factible con $|Q|+1$ desigualdades.

3.2. DOBSS-SSG

En este caso, notar que la función objetivo de 3.7, también puede expresarse en función de 3.8 de la siguiente manera:

$$\sum_{l \in L} \sum_{j \in Q} \sum_{i \in X} p^l \cdot R_{ij}^l \cdot z_{ij}^l = \sum_{l \in L} p^l \cdot \sum_{j \in Q} (Rd_j^l \cdot \sum_{i \in X: j \in i} z_{ij}^l + Pd_j^l \cdot \sum_{i \in X: j \notin i} z_{ij}^l)$$

Para poder obtener una representación concisa que no use las variables X, se debe hallar una forma concisa de representar $\sum_{i \in X: j \in i} z_{ij}^l$ y para ello se define:

$$\hat{w}_j^l = \sum_{i \in X: j \in i} z_{ij}^l \quad (3.14)$$

$$w_{jk}^l = \sum_{i \in X: j \in i} z_{ik}^l \quad (3.15)$$

Donde la variable \hat{w}_j^l corresponde a la suma de z_{ij}^l para todas las estrategias i que protegen j y la variable w_{jk}^l es la suma de z_{ik}^l para todas las estrategias i que protegen k . Esto significa que $\hat{w}_j^l = w_{jj}^l$. Usando la variable w_{jk}^l y la restricción (3.7b) tenemos que:

$$w_{jj}^l + \sum_{i \in X: j \notin i} z_{ij}^l = q_j^l \quad \forall j \in Q, \forall l \in L$$

Con este término se puede aprovechar la estructura del caso de seguridad mostrada en (3.8) para reescribir la función objetivo del MILP (3.7) de la siguiente manera:

$$\begin{aligned}
\sum_{l \in L} \sum_{j \in Q} \sum_{i \in X} p^l \cdot R_{ij}^l \cdot z_{ij}^l &= \sum_{l \in L} p^l \cdot \sum_{j \in Q} (Rd_j^l \cdot \sum_{i \in X: j \in i} z_{ij}^l + Pd_j^l \cdot \sum_{i \in X: j \notin i} z_{ij}^l) \\
&= \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l \cdot \hat{w}_j^l + Pd_j^l \cdot (q_j^l - \hat{w}_j^l)) \\
&= \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot \hat{w}_j^l + p^l \cdot Pd_j^l \cdot q_j^l \\
&= \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l
\end{aligned} \tag{3.16}$$

De forma análoga, se reescribe la restricción (3.7h) usando las variables w_{jk}^l dada en (3.15) junto con la definición de y dada en (3.9) y considerando la restricción (3.7c) tenemos lo siguiente:

$$\begin{aligned}
\sum_{l \in L} \sum_{k \in Q} C_{ij}^l \cdot z_{ij}^l &= \sum_{k \in Q} (\sum_{i \in X: j \in i} Pa_j^l \cdot z_{ik}^l + \sum_{i \in X: j \notin i} Ra_j^l \cdot z_{ij}^l) \\
&= \sum_{k \in Q} (Pa_j^l \cdot w_{jk}^l + Ra_j^l \cdot (q_k^l - w_{jk}^l)) \\
&= Pa_j^l \cdot y_j + Ra_j^l \cdot (1 - y_j)
\end{aligned} \tag{3.17}$$

Finalmente usando las transformaciones (3.16) y (3.17) junto con restricciones lógicas que relacionen las variables w , q e y , obtenemos el segundo modelo denominado (DOBSS-SSG), formulación (3.18).

$$\begin{aligned}
Max_{w,q} \quad & \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\
s.t. \quad & \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.18a) \\
& q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.18b) \\
& \sum_{j \in Q} y_j = m \quad (3.18c) \\
& 0 \leq y \leq 1 \quad (3.18d) \\
& 0 \leq w_{jk}^l \leq q_k^l \quad \forall l \in L, \forall j, k \in Q \quad (3.18e) \\
& y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.18f) \\
& 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.18g)
\end{aligned}$$

Notar que (DOBSS-SSG) resuelve para un vector óptimo de probabilidades de protección de un objetivo y el vector óptimo de ataque para el atacante. Recuperar la estrategia mixta óptima para el defensor en base a las estrategias puras originales puede ser hecho en tiempo polinomial.

3.3. MIP-SSG

Esta tercera formulación no usa restricciones del tipo M grande. Esta formulación es idéntica al MILP (3.7) con la excepción que la restricción asociada a la mejor respuesta del atacante no usa la gran M, es decir, en lugar de usar la restricción (3.7h) usa:

$$\sum_{i \in X} (C_{ij}^l - C_{ik}^l) \cdot z_{ij}^l \geq 0, \quad \forall l \in L, \forall j, k \in Q \quad (3.19)$$

La restricción (3.19) también puede reescribirse aprovechando la estructura del caso de seguridad mostrado en (3.8) y usando las mismas variables \hat{w} y w definidas en (3.14) y (3.15):

$$\begin{aligned}
\sum_{i \in X} (C_{ij}^l - C_{ik}^l) z_{ij}^l &= \sum_{i \in X} C_{ij}^l z_{ij}^l - \sum_{i \in X} C_{ik}^l z_{ij}^l \\
&= \sum_{i \in X: j \in i} Pa_j^l z_{ij}^l + \sum_{i \in X: j \notin i} Ra_j^l z_{ij}^l - \sum_{i \in X: k \in i} Pa_k^l z_{ij}^l - \sum_{i \in X: k \notin i} Ra_k^l z_{ij}^l \\
&= Pa_j^l \sum_{i \in X: j \in i} z_{ij}^l + Ra_j^l \sum_{i \in X: j \notin i} z_{ij}^l - Pa_k^l \sum_{i \in X: k \in i} z_{ij}^l - Ra_k^l \sum_{i \in X: k \notin i} z_{ij}^l \\
&= Pa_j^l \hat{w}_j^l + Ra_j^l (q_j^l - \hat{w}_j^l) - Pa_k^l w_{jk}^l - Ra_k^l (q_k^l - w_{jk}^l) \\
&= Pa_j^l w_{jj}^l + Ra_j^l (q_j^l - w_{jj}^l) - Pa_k^l w_{jk}^l - Ra_k^l (q_k^l - w_{jk}^l) \\
&= (Pa_j^l - Ra_j^l) w_{jj}^l + Ra_j^l q_j^l - (Pa_k^l - Ra_k^l) w_{jk}^l - Ra_k^l q_k^l \tag{3.20}
\end{aligned}$$

Entonces, usando el MILP (3.18), reemplazando la restricción (3.18g) por la restricción (3.19) expresándala en los términos obtenidos en (3.20), obtenemos el tercer modelo al cual hemos denominado (MIP-SSG), formulación (3.21).

$$\begin{aligned}
Max_{w,q} \quad & \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\
s.t. \quad & \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \tag{3.21a} \\
& q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \tag{3.21b} \\
& \sum_{j \in Q} y_j = m \tag{3.21c} \\
& 0 \leq y \leq 1 \tag{3.21d} \\
& 0 \leq w_{jk}^l \leq q_k^l \quad \forall l \in L, \forall j, k \in Q \tag{3.21e} \\
& y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l \quad \forall l \in L, \forall j, k \in Q \tag{3.21f} \\
& (Pa_j^l - Ra_j^l) \cdot w_{jj}^l + Ra_j^l \cdot q_j^l \geq \\
& (Pa_k^l - Ra_k^l) \cdot w_{kj}^l + Ra_k^l \cdot q_j^l \quad \forall l \in L, \forall j, k \in Q \tag{3.21g}
\end{aligned}$$

Notar que (MIP-SSG) resuelve para un vector óptimo de probabilidades de protección de un objetivo y el vector óptimo de ataque para el atacante. Recuperar la estrategia mixta óptima para el defensor en base a las estrategias puras originales puede ser hecho en tiempo polinomial.

3.4. Formulaciones Logarítmicas

En esta sección revisamos una técnica que permite reducir el número de variables binarias a $\lceil \log_2(k) \rceil$, donde k corresponde al número de objetivos a proteger. Una forma de construir formulaciones MIP con un número logarítmico de variables es usar la siguiente proposición de [41]:

Proposición 3.1 Sean $\{b^i\}_{i=1}^k \subseteq \{0, 1\}^{\lceil \log_2(k) \rceil}$ un conjunto de k vectores binarios de dimensión $\lceil \log_2(k) \rceil$, tales que $b^i \neq b^j \quad \forall i, j \in \{1 \dots k\}$ con $i \neq j$. Entonces el conjunto $S = \{y \in \{0, 1\}^k \mid \sum_{i=1}^k y_i = 1\}$ es equivalente a:

$$\begin{aligned}
 \sum_{i=1}^k y^i &= 1 \\
 \sum_{i=1}^k b^i \cdot y^i &= w \\
 w &\in \{0, 1\}^{\lceil \log_2(k) \rceil} \\
 y^i &\geq 0 \quad \forall i \in \{1 \dots k\}
 \end{aligned} \tag{3.22}$$

Recientemente, formulaciones idénticas a esta han sido propuestas de forma independiente en [1], [16], [28] y [42]; sin embargo, la idea detrás de esta proposición es parte de la programación matemática desde hace mucho tiempo, ver [39].

Este trabajo de tesis utiliza la proposición (3.1) sobre el conjunto $S = \{q^l \in \{0, 1\}^{|Q|} \mid \sum_{j \in Q} q_j^l = 1, \quad \forall l \in L\}$, es decir, (3.22) se reescribe como (3.23) donde los vectores $\{b^j\} \subseteq \{0, 1\}^{\lceil \log_2(|Q|) \rceil}$ son tales que cada vector en $\{0, 1\}^{\lceil \log_2(|Q|) \rceil}$ corresponden a una rama de un árbol binario con $\lceil \log_2(|Q|) \rceil$ niveles tales que cada componente corresponde a un nivel y el valor del componente indica si la rama seleccionada se encuentra en dicho nivel.

$$\begin{aligned}
 \sum_{j \in Q} q_j^l &= 1 \quad \forall l \in L \\
 \sum_{j \in Q} b^j \cdot q_j^l &= w^l \quad \forall l \in L \\
 w &\in \{0, 1\}^{\lceil \log_2(|Q|) \rceil} \\
 q_j^l &\geq 0 \quad \forall j \in |Q|
 \end{aligned} \tag{3.23}$$

3.4.1. LSSGM

Este modelo usa la proposición (3.1) en el MILP (3.13), reemplazando las restricciones (3.13f) y (3.13e) por (3.23) se obtiene la primera formulación logarítmica a la cual se denomina (LSSGM), esta se detalla en el MILP (3.24).

$$\begin{aligned}
 &Max_{q,a,d} \quad \sum_{l \in L} p^l \cdot d^l \\
 &s.t. \quad d^l \leq Rd_j^l \cdot y_j + Pd_j^l \cdot (1 - y_j) + M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.24a) \\
 &\quad 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.24b) \\
 &\quad \sum_{j \in Q} y_j = m \quad (3.24c) \\
 &\quad 0 \leq y \leq 1 \quad (3.24d) \\
 &\quad \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.24e) \\
 &\quad q_j^l \geq 0 \quad \forall l \in L, \forall j \in Q \quad (3.24f) \\
 &\quad \sum_{j \in Q} b^j \cdot q_j^l = w \quad \forall l \in L \quad (3.24g) \\
 &\quad w^l \in \{0, 1\}^{\log_2(|Q|)} \quad \forall l \in L \quad (3.24h)
 \end{aligned}$$

3.4.2. LDOBSS-SSG

Este modelo usa la proposición (3.1) en el MILP (3.18), reemplazando las restricciones (3.18b) y (3.18a) por (3.23) se obtiene la segunda formulación logarítmica a la cual se denomina (LDOBSS-SSG), esta se detalla en el MILP (3.25).

$$\begin{aligned}
 &Max_{w,q} \quad \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\
 &s.t. \quad \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.25a) \\
 &\quad \sum_{j \in Q} b^j \cdot q_j^l = w^l \quad \forall l \in L \quad (3.25b) \\
 &\quad w^l \in \{0, 1\}^{\lceil \log_2(|Q|) \rceil} \quad \forall l \in L \quad (3.25c) \\
 &\quad q_j^l \geq 0 \quad \forall l \in L, \forall j \in Q \quad (3.25d) \\
 &\quad \sum_{j \in Q} y_j = m \quad (3.25e) \\
 &\quad 0 \leq y \leq 1 \quad (3.25f)
 \end{aligned}$$

$$0 \leq w_{jk}^l \leq q_k^l \quad \forall l \in L, \forall j, k \in Q \quad (3.25g)$$

$$y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.25h)$$

$$0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.25i)$$

3.4.3. LMIP-SSG

Este modelo usa la proposición (3.1) en el MILP (3.21), reemplazando las restricciones (3.21b) y (3.21a) por (3.23) se obtiene la tercera formulación logarítmica a la cual se denomina (LMIP-SSG), esta se detalla en el MILP (3.26).

$$Max_{w,q} \quad \sum_{l \in L} \sum_{j \in J} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l$$

$$s.t. \quad \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.26a)$$

$$\sum_{j \in Q} b^j \cdot q_j^l = w^l \quad \forall l \in L \quad (3.26b)$$

$$w^l \in \{0, 1\}^{\lceil \log_2(|Q|) \rceil} \quad \forall l \in L \quad (3.26c)$$

$$q_j^l \geq 0 \quad \forall l \in L, \forall j \in Q \quad (3.26d)$$

$$\sum_{j \in Q} y_j = m \quad (3.26e)$$

$$0 \leq y \leq 1 \quad (3.26f)$$

$$0 \leq w_{jk}^l \leq q_k^l \quad \forall l \in L, \forall j, k \in Q \quad (3.26g)$$

$$y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.26h)$$

$$(Pa_j^l - Ra_j^l) \cdot w_{jj}^l + Ra_j^l \cdot q_j^l \geq$$

$$(Pa_k^l - Ra_k^l) \cdot w_{kj}^l + Ra_k^l \cdot q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.26i)$$

3.5. Incorporando restricciones al Espacio de Soluciones

Hay muchas posibles extensiones a las primeras tres formulaciones SSG. Dichas extensiones son, por lo general, importantes para implementar aplicaciones en ámbitos diferentes. Una de estas posibles extensiones esta relacionada al espacio de soluciones del defensor.

En todos los modelos descritos en las secciones anteriores se considera que todas las posibles combinaciones de asignaciones de recursos a objetivos son factibles. Sin embargo,

en las aplicaciones del mundo real esto no siempre es posible, es más, por lo general, es todo lo contrario.

Un claro ejemplo es el caso del Servicio Federal de Alguaciles Aéreos (FAMS) donde no todas las combinaciones de alguaciles a aviones son factibles ya que para programar un alguacil a un vuelo, este tiene que estar previamente en la locación de origen y se debe considerar el tiempo de vuelo entre locaciones antes de dar una siguiente asignación a un recurso [40]. En este trabajo, redefinen la noción de los objetivos para que en vez de considerar cada tramo de vuelo como un objetivo consideran a pares tramos factibles.

En esta sección se discute dos posibles restricciones a las acciones de defensa del defensor, considerando que no todas las combinaciones de asignación de recursos a objetivos son factibles. La primera usa la definición clásica de adyacencia permitiendo asignaciones sólo a objetivos consecutivos de tamaño m , con m el número de recursos de seguridad disponibles; y la segunda, utiliza la noción de locaciones prohibidas en una misma estrategia, permitiendo asignaciones sólo a estrategias que contengan locaciones sin impedimentos entre sí.

3.5.1. Restricción a objetivos consecutivos

Es ampliamente reconocido que las condiciones de adyacencia pueden ser interpretadas de varias maneras, en general, depende de la noción de proximidad concebida para un cierto ámbito, la cual puede ser espacial como no-espacial. El enfoque clásico se refiere a simplemente interpretar adyacencia cuando dos locaciones comparten un borde o punto en común y por ello este enfoque es también conocido como adyacencia de puntos colindantes. Un enfoque alternativo de adyacencia es definir dos locaciones como adyacentes siempre que estas estén a distancias similares de un punto de referencia o que tengan una distancia similar entre ambas.

En esta sección se trabaja el término de locaciones consecutivas bajo esta definición alterna de adyacencia, motivados por el problema de los guardabosques como el descrito en [12], donde los agentes encargados de prevenir la caza furtiva de animales deben tomar en consideración la elevación de los terrenos y la distancia entre el punto de vigilancia y el centro de operaciones o campo base desde donde salen a patrullar. De esta forma se reparte un esfuerzo equitativo entre los agentes de un mismo turno. Teniendo estos conceptos en mente, se considera que los m recursos de seguridad del defensor, deben ser asignados a objetivos consecutivos y se define la variable y como (3.27).

$$y_j = \sum_{i \in \max\{1, j-m+1\}}^{\min\{j, N-m+1\}} x_i, \quad \forall j \in Q \quad (3.27)$$

Utilizando (3.27) en las formulaciones (3.13), (3.18) y (3.21) se obtiene los modelos (3.28), (3.29) y (3.30), correspondientes a locaciones consecutivas.

En estas tres formulaciones, el espacio de solución del defensor queda reducido a un tamaño de $|X| = Q - m + 1$.

$$\begin{aligned}
Max_{q,a,d} \quad & \sum_{l \in L} p^l \cdot d^l \\
s.t. \quad & d^l \leq Rd_j^l \cdot y_j + Pd_j^l \cdot (1 - y_j) + M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.28a) \\
& 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.28b) \\
& \sum_{j \in Q} y_j = m \quad (3.28c) \\
& 0 \leq y \leq 1 \quad (3.28d) \\
& \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.28e) \\
& q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.28f) \\
& \sum_{i \in X} x_i = 1 \quad (3.28g) \\
& x_i \geq 0 \quad \forall i \in X \quad (3.28h) \\
& y_j = \sum_{i \in \max\{1, j-m+1\}}^{\min\{j, N-m+1\}} X_i \quad \forall j \in Q \quad (3.28i)
\end{aligned}$$

$$\begin{aligned}
Max_{w,q} \quad & \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\
s.t. \quad & \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.29a) \\
& q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.29b) \\
& \sum_{j \in Q} y_j = m \quad (3.29c) \\
& 0 \leq y \leq 1 \quad (3.29d) \\
& 0 \leq w_{jk}^l \leq q_k^l \quad \forall l \in L, \forall j, k \in Q \quad (3.29e) \\
& y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.29f)
\end{aligned}$$

$$0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.29g)$$

$$\sum_{i \in X} x_i = 1 \quad (3.29h)$$

$$x_i \geq 0 \quad \forall i \in X \quad (3.29i)$$

$$y_j = \sum_{i \in \max\{1, j-m+1\}}^{\min\{j, N-m+1\}} X_i \quad \forall j \in Q \quad (3.29j)$$

$$\begin{aligned} \text{Max}_{w,q} \quad & \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\ \text{s.t.} \quad & \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.30a) \end{aligned}$$

$$q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.30b)$$

$$\sum_{j \in Q} y_j = m \quad (3.30c)$$

$$0 \leq y \leq 1 \quad (3.30d)$$

$$0 \leq w_{jk}^l \leq q_k^l \quad \forall l \in L, \forall j, k \in Q \quad (3.30e)$$

$$y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.30f)$$

$$\begin{aligned} & (Pa_j^l - Ra_j^l) \cdot w_{jj}^l + Ra_j^l \cdot q_j^l \geq \\ & (Pa_k^l - Ra_k^l) \cdot w_{kj}^l + Ra_k^l \cdot q_j^l \quad \forall l \in L, \forall j, k \in Q \quad (3.30g) \end{aligned}$$

$$\sum_{i \in X} x_i = 1 \quad (3.30h)$$

$$x_i \geq 0 \quad \forall i \in X \quad (3.30i)$$

$$y_j = \sum_{i \in \max\{1, j-m+1\}}^{\min\{j, N-m+1\}} x_i \quad \forall j \in Q \quad (3.30j)$$

3.5.2. Restricción de objetivos en una misma estrategia

En el ámbito de los SSGs, pueden existir escenarios donde una cierta cantidad de recursos no pueda ser asignada junto a otro u otros objetivos por motivos operacionales o limitaciones de acceso.

Este escenario esta motivado por el problema de los guardabosques descrito en [13], donde los agentes encargados de prevenir la caza furtiva de zebras deben tomar en consideración los tiempos de viaje para llegar a zonas típicas de distribución de los animales, algunas zonas pueden ser accedidas en menos de 6 horas mientras que otras tienen un tiempo de viaje de más de 35 horas de camino. Estas limitaciones de acceso dan la noción de zonas "prohibidas"

de ser asignadas junto a otras locaciones al mismo tiempo por temas logísticos, limitando de esta manera el espacio de estrategias del defensor.

Definiendo esta característica formalmente, sea $I \subset Q$ el conjunto de objetivos que tiene alguna incompatibilidad con otro objetivo y sea $P(j)$ el conjunto de objetivos prohibidos del objetivo $j \in I$ y denominando y al vector de protección de un objetivo que suma todas las estrategias puras que protegen dicho objetivo, y queda definida como (3.31).

$$y_j = \sum_{i \in X: i \notin P(j)} x_i, \quad \forall j \in Q \quad (3.31)$$

Utilizando (3.31) en las formulaciones (3.13), (3.18) y (3.21), se obtiene las formulaciones de interés para este caso (3.32), (3.33) y (3.34).

$$\begin{aligned} \text{Max}_{q,a,d} \quad & \sum_{l \in L} p^l \cdot d^l \\ \text{s.t.} \quad & d^l \leq Rd_j^l \cdot y_j + Pd_j^l \cdot (1 - y_j) + M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.32a) \\ & 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) \quad \forall l \in L, \forall j \in Q \quad (3.32b) \\ & \sum_{j \in Q} y_j = m \quad (3.32c) \\ & 0 \leq y \leq 1 \quad (3.32d) \\ & \sum_{j \in Q} q_j^l = 1 \quad \forall l \in L \quad (3.32e) \\ & q^l \in \{0, 1\}^{|Q|} \quad \forall l \in L \quad (3.32f) \\ & \sum_{i \in X} x_i = 1 \quad (3.32g) \\ & x_i \geq 0 \quad \forall i \in X \quad (3.32h) \\ & y_j = \sum_{i \in X: i \notin P(j)} x_i, \quad \forall j \in Q \quad (3.32i) \end{aligned}$$

$$\begin{aligned}
Max_{w,q} \quad & \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\
s.t. \quad & \sum_{j \in Q} q_j^l = 1 & \forall l \in L & (3.33a) \\
& q^l \in \{0, 1\}^{|Q|} & \forall l \in L & (3.33b) \\
& \sum_{j \in Q} y_j = m & & (3.33c) \\
& 0 \leq y \leq 1 & & (3.33d) \\
& 0 \leq w_{jk}^l \leq q_k^l & \forall l \in L, \forall j, k \in Q & (3.33e) \\
& y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l & \forall l \in L, \forall j, k \in Q & (3.33f) \\
& 0 \leq a^l - Pa_j^l \cdot y_j - Ra_j^l \cdot (1 - y_j) \leq M \cdot (1 - q_j^l) & \forall l \in L, \forall j \in Q & (3.33g) \\
& \sum_{i \in X} x_i = 1 & & (3.33h) \\
& x_i \geq 0 & \forall i \in X & (3.33i) \\
& y_j = \sum_{i \in X: i \notin P(j)} x_i, & \forall j \in Q & (3.33j)
\end{aligned}$$

$$\begin{aligned}
Max_{w,q} \quad & \sum_{l \in L} \sum_{j \in Q} p^l \cdot (Rd_j^l - Pd_j^l) \cdot w_{jj}^l + p^l \cdot Pd_j^l \cdot q_j^l \\
s.t. \quad & \sum_{j \in Q} q_j^l = 1 & \forall l \in L & (3.34a) \\
& q^l \in \{0, 1\}^{|Q|} & \forall l \in L & (3.34b) \\
& \sum_{j \in Q} y_j = m & & (3.34c) \\
& 0 \leq y \leq 1 & & (3.34d) \\
& 0 \leq w_{jk}^l \leq q_k^l & \forall l \in L, \forall j, k \in Q & (3.34e) \\
& y_j - 1 + q_j^l \leq w_{jk}^l \leq y_j + 1 - q_j^l & \forall l \in L, \forall j, k \in Q & (3.34f) \\
& (Pa_j^l - Ra_j^l) \cdot w_{jj}^l + Ra_j^l \cdot q_j^l \geq & & \\
& (Pa_k^l - Ra_k^l) \cdot w_{kj}^l + Ra_k^l \cdot q_j^l & \forall l \in L, \forall j, k \in Q & (3.34g) \\
& \sum_{i \in X} x_i = 1 & & (3.34h) \\
& x_i \geq 0 & \forall i \in X & (3.34i) \\
& y_j = \sum_{i \in X: i \notin P(j)} x_i, & \forall j \in Q & (3.34j)
\end{aligned}$$

Capítulo 4

Estudio Computacional

4.1. Introducción

Con el fin de comparar el rendimiento de las diferentes formulaciones de programación lineal entera mixta, se realizaron experimentos con instancias aleatorias, descritas en detalle más adelante.

El objetivo de estos experimentos es evaluar:

1. El tamaño de las diferentes formulaciones. Enfocado al número de columnas, filas y no-ceros. Se da una especial atención a los cambios en la estructura de la matriz de recompensa.
2. La fortaleza de la cota dada por la relajación lineal. Se compara la proximidad del límite dado por la relajación lineal con el valor de la mejor solución factible conocida.
3. El rendimiento de las formulaciones. Se analiza la cantidad de tiempo que se requiere para resolver los problemas con cada formulación. Se compara principalmente la cantidad de instancias que llegan a optimalidad dentro del tiempo límite establecido y el gap de las instancias que no logran resolver en dicho tiempo.
4. El efecto que tiene la representación logarítmica de los targets en las formulaciones y su escalabilidad.
5. El efecto que tiene restringir el espacio de soluciones del defensor en las formulaciones.

Las instancias a evaluar son generadas de manera aleatoria, sus matrices de utilidades tienen valores de pérdida entre $[-20, 0]$ y de recompensa entre $[0, 20]$, con excepción del 10 % de las entradas que tienen un orden de magnitud mayor en todos los experimentos, al menos se especifique lo contrario en alguna sección. Se usa este tipo de configuración porque se considera que existen objetivos de mayor importancia con respecto a los demás.

Las utilidades de defensor y los atacantes cumplen lo especificado al inicio del capítulo 3. En aquellas formulaciones donde se usa la técnica de la M grande, el valor de esta se determina multiplicando el coeficiente más grande de la matriz por dos.

El primer set de instancias, set A, corresponde a instancias de dificultad baja y media. El set B, denominado Caso Base, corresponde a una dificultad media. Mientras que los sets C y D corresponden a instancias de dificultad alta. Los sets de instancias B, C y D fueron elegidos y caracterizados de esa manera porque sus parámetros hacen inducir una mayor complejidad de resolución. Considerar que se les da esa caracterización de dificultad por las observaciones hechas en una etapa exploratoria preliminar donde, por ejemplo, aquellas instancias con un número bajo de atacantes, 2 o 3 tipos de atacantes, y con un número bajo de objetivos, 10 objetivos o menos, el tiempo de resolución es de menos de 5 segundos. A estas instancias se las caracteriza como de dificultad baja, y conforme aumenta el valor de los parámetros de las formulaciones se les caracteriza con una dificultad mayor.

Los resultados computacionales, por ende, deben ser considerados ilustrativos como tipos de problemas para el problema de asignación de recursos de seguridad.

Cada set de instancias contiene un rango de valores para cada uno de los parámetros que nos interesa evaluar. En la tabla 4.1 se resume la información básica concerniente a estas instancias.

Instancia	Tipos de Atacantes L	Targets Q	Recursos de Defensa m
A	[2, 4, 6, 8]	[10, 20, 30, 40, 50]	[0,3 Q , 0,4 Q , 0,5 Q , 0,6 Q , 0,7 Q]
B	6	40	0,3 Q
C	[2...9]	40	0,3 Q
D	2	[64, 256, 512, 1024]	0,3 Q

Tabla 4.1: Información de las Instancias

Todos los experimentos fueron realizados en una máquina con sistema operativo Linux 2.6.32 con 16Gb de RAM y dos Quad-Core Intel Xeon E5420. Todos los programas fueron escritos en el lenguaje de programación Python 2.7, y Gurobi 6.0.4 para resolver los programas lineales y enteros. Todas las ejecuciones se realizaron con las configuraciones predeterminadas y, al menos se indique lo contrario, se impuso un tiempo límite de 10 minutos.

Finalmente, tomar nota que los tiempos reportados no toman en consideración el tiempo que toma construir la instancia, es decir, el tiempo que se requiere para generar las matrices aleatorias; sin embargo, esto se ejecuta rápidamente con un tiempo promedio de construcción de 60 segundos.

4.2. Tamaño de las diferentes formulaciones

El primer experimento consistió en comparar los tamaños resultantes de las tres formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), con un tiempo límite de 600 segundos, realizando 20 ejecuciones por cada set de parámetros L y Q , y usa una matriz generada aleatoriamente con una estructura que considera el 10 % de los datos con un orden de magnitud de 10^2 , un orden más que el resto de datos, elegidos en posiciones aleatorias. Las razones de esta configuración se especifican al inicio de este capítulo y se generan varias ejecuciones de manera aleatoria para generalizar.

En la Tabla 4.2 se describe el número de variables (Var), restricciones (Restr) y coeficientes no-cero (NZ) para cada una de las tres formulaciones para el set de instancias A, cuando esta toma 40 objetivos y un ratio de recursos de defensa de $(0,3 \cdot |Q|)$.

	SSGM			DOBSSG			MIPSSG		
$ L $	Var	Restr	NZ	Var	Restr	NZ	Var	Restr	NZ
2	124	243	760	3322	9763	26120	3320	12803	34952
4	208	485	1479	6604	19525	52200	6600	25605	69859
6	292	727	2198	9886	29287	78279	9880	38407	104730
8	376	969	2919	13168	39049	104359	13160	51209	139700

Tabla 4.2: Tamaño del set A cuando $|N| = 40$, $m = (0,3 \cdot |Q|)$ y $|Q|$ varía.

Por ejemplo, consideremos las instancias cuando $|L| = 4$. Se puede ver que el número total de filas y columnas de la formulación (SSGM) es $208 + 485 = 693$ y para (DOBSS-SSG) es $6604 + 19525 = 13064$ y por otro lado se puede ver que la formulación (MIP-SSG) resulta en problemas mucho más grandes, en este caso con $6600 + 25605 = 32205$. Es decir, (MIP-SSG) es 4547 % más grande que (SSGM) y 23 % más grande que (DOBSS-SSG).

En general, si se considera todas las instancias del set A, la formulación (MIP-SSG) tiene en promedio un tamaño 4602 % más grande que (SSGM) en términos de número de filas y columnas y, (MIP-SSG) es también la formulación más densa en términos de no-ceros, llegando a ser 4648 % más densa que (SSGM) y 34 % más densa que (DOBSS-SSG).

4.3. Fortaleza de la relajación lineal

El segundo experimento consistió en resolver la relajación lineal usando las instancias del set C, ver parámetros en la tabla 4.1, para las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), ver modelos (3.13), (3.18) y (3.21) respectivamente.

El diseño de este experimento considera un tiempo límite de 600 segundos, 50 ejecuciones

por variación de L , usa una matriz generada aleatoriamente con una estructura que considera un 10 % de los datos de un orden de magnitud de 10^2 , un orden mayor al resto, elegidos en posiciones aleatorias.

Para cada instancia se registró el tiempo que tomó en resolver el problema (tiempo del LP en la raíz) y adicionalmente el valor óptimo de cada instancia. Para cada instancia se compara el valor de la relajación lineal con el valor óptimo de la instancia correspondiente para obtener el gap del LP en la raíz. Si z^* es el valor de la mejor solución conocida para una instancia y zr es el valor de la relajación lineal obtenida para una formulación, entonces el valor del gap en la raíz del LP sería $(\frac{zr}{z^*} - 1) \cdot 100$.

Por ejemplo, si la mejor solución óptima para la instancia 1 es $z^* = 10,0$ y el valor de la relajación lineal de la formulación SSGM es $zr = 12,0$; entonces el valor del gap del lp en la raíz es: $(\frac{104,2}{100,0} - 1) \cdot 100 = 4,2$.

Los resultados se muestran en la Tabla 4.3

L	SSGM		DOBSS-SSG		MIP-SSG	
	Tiempo(s)	LP Gap(%)	Tiempo(s)	LP Gap(%)	Tiempo(s)	LP Gap(%)
2	0.0032	698.8	0.0400	287.2	0.0888	156.0
3	0.0046	1042.8	0.0840	448.1	0.1294	252.5
4	0.0053	1197.7	0.1036	543.3	0.1884	308.7
5	0.0065	1455.3	0.1264	681.1	0.2566	399.9
6	0.0070	1791.7	0.1486	890.0	0.3215	536.0
7	0.0078	1916.1	0.1787	965.2	0.3875	583.4
8	0.0099	2159.7	0.1996	1131.6	0.4743	673.4
9	0.0109	2161.3	0.2334	1165.0	0.5646	704.1

Tabla 4.3: Fortaleza de la Relajación Lineal y Tiempo de ejecución del LP para el Set C

Surgen las siguientes observaciones:

Se puede ver que la formulación (MIP-SSG) tiene el gap más ajustado del LP en el nodo raíz. Por ejemplo, consideremos el caso de $|L| = 6$. El gap en el nodo raíz para la formulación (SSGM) es 1791 % y para (DOBSS-SSG) es 890 %; mientras que para (MIP-SSG) es 536 %. Sin embargo, en general las tres formulaciones tienen un pésimo gap en el nodo raíz. Si se considera todas las ejecuciones, (SSGM), (DOBSS-SSG) y (MIP-SSG) tienen un gap promedio de 1552.9 %, 763.9 % y 451.8 % respectivamente.

Aunque hay que tomar en cuenta que, considerando todas las ejecuciones, el tiempo promedio de resolución de la relajación lineal de las tres formulaciones es muy rápido, toma aproximadamente 1 segundo o menos, ver cuadro 4.3. Tomar nota que el tiempo requerido para resolver la relajación lineal de la formulación (SSGM) es considerablemente menor que el requerido para resolver las formulaciones (DOBSS-SSG) y (MIP-SSG).

4.4. Rendimiento de las formulaciones

El objetivo de esta sección es determinar el rendimiento de las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

La principal herramienta para medir el rendimiento es el porcentaje acumulado de instancias resueltas dentro del tiempo límite. También se analiza el gap en aquellos sets donde no se pudo llegar a optimalidad dentro del tiempo establecido. Dada una ejecución del algoritmo de branch and bound, sea zu el valor actual del cota superior y zl el valor de la mejor solución factible conocida hasta el momento. El gap se calcula así: $(\frac{zu}{zl} - 1) \cdot 100$. Finalmente, se identifica la cantidad de nodos de Branch and Bound que se generan en cada una de las formulaciones.

Este experimento utiliza los set de instancias A y B detallados en el cuadro 4.1. El set de instancias A se usa para determinar un perfil inicial de rendimiento de las formulaciones y su análisis se detalla en la sección 4.4.1. El set B se usa como caso base para determinar la sensibilidad ante cambios en los parámetros y su análisis se detalla en las secciones 4.4.2, 4.4.3, 4.4.4, 4.4.5 y 4.4.6.

4.4.1. Tiempos de resolución

El propósito de esta sección es determinar un perfil para las tres formulaciones ya mencionadas. Para ello, se usa el set A, ver parámetros en la tabla 4.1. El diseño de este experimento considera un tiempo límite de 600 segundos, 20 ejecuciones por cada set de parámetros L y Q , y usa una matriz generada aleatoriamente con una estructura que considera un 10 % de los datos con un orden de magnitud de 10^2 , un orden mayor que el resto, elegidos en posiciones aleatorias.

Los resultados obtenidos para el set de instancias A se muestran en la figura 4.1

Aquí surgen distintas observaciones:

La formulación (SSGM) se comporta mejor en un 80 % de instancias, que justamente corresponde a las instancias de menor dificultad, ver parte (a) de la figura 4.1.

En el 20 % de instancias restantes, la formulación (DOBSS-SSG) domina ligeramente sobre (MIP-SSG) pero en general ambas tienen un rendimiento similar. Estas son instancias de mediana dificultad por lo que el enfoque del análisis en las secciones posteriores se centra en dificultad media-alta para determinar si una de estas dos formulaciones se comporta mejor que la otra y bajo qué parámetros.

Las tres formulaciones resuelven entre el 10 % y 20 % de instancias en el nodo raíz y no

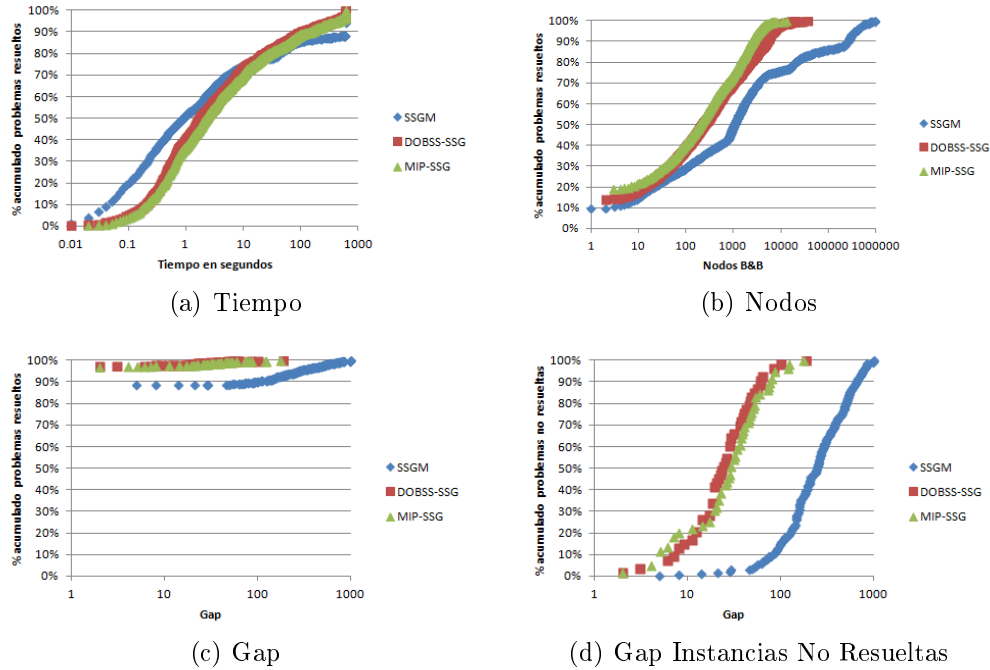


Figura 4.1: Set A: Tiempo, Nodos y Gap vs % Acumulado de Problemas Resueltos.

requieren ramificar el árbol, esto se muestra en la parte (b) de la figura 4.1. Aquí también se puede apreciar que la formulación (SSGM) es la que implica la menor cantidad de nodos y es la única que presenta un cambio en su comportamiento, a partir del 40 % de instancias el número de nodos aumenta considerablemente y esto justo corresponde a instancias con $|L| \geq 6$.

Las instancias que no lograron encontrar una solución óptima dentro del tiempo establecido son apenas el 10 % del total de ejecuciones, como se puede apreciar en la parte (c) de la figura 4.1.

El gap obtenido en las instancias que no alcanzan optimalidad queda cerca o bajo el 100 % de gap en las formulaciones (DOBSS-SSG) y (MIP-SSG); mientras que en la formulación (SSGM) sobrepasa el 100 % de gap en un 80 % de las instancias no resueltas a optimalidad, ver parte (d) de la figura 4.1

La tabla 4.4 muestra los resultados del set de instancias A cuando $|L| = 4$, $m = (0,3|Q|)$ y $|Q|$ varía. Aquí se puede apreciar que estas son instancias de baja dificultad, todas se resuelven a optimalidad en menos de 30 segundos en promedio. En este abstracto del set A, se puede ver que la formulación (SSGM) es la ganadora hasta que la cantidad de objetivos sube a $|Q| = 50$ donde (DOBSS-SSG) empieza a ganar. Esta tabla se muestra solo a modo de detalle de los resultados mostrados en la figura 4.1.

$ Q $	SSGM	DOBSS-SSG	MIP-SSG
10	0.06	0.27	0.19
20	0.21	0.65	0.82
30	0.59	1.89	2.68
40	1.90	4.48	6.56
50	27.70	9.26	15.19

Tabla 4.4: Tiempo de Ejecución del set A cuando $|L| = 4$, $m = (0,3 | Q |)$ y $|Q|$ varía

Para entender mejor el comportamiento de las formulaciones es necesario profundizar el detalle del análisis. Por ello, se toma un caso base a partir de las observaciones del set de instancias A. Los parámetros de este caso base, se pueden ver bajo el nombre de set B en el cuadro 4.1 y su análisis se detalla a continuación.

4.4.2. Efecto del orden de magnitud de las entradas de la matriz de utilidades

El propósito de esta sección es determinar si el orden de magnitud del 10 % de datos influye en el rendimiento de las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Para ello, se toma como caso base el set B, ver parámetros en la tabla 4.1. El diseño de este experimento considera un tiempo límite de 600 segundos, 50 ejecuciones por orden de magnitud y usa una matriz generada aleatoriamente con el 10 % de los datos de mayor magnitud elegidos en posiciones aleatorias. El orden de magnitud de este 10 % de datos parte en 10^2 y sube un orden de magnitud a la vez hasta llegar a 10^5 .

Los resultados se muestran en la figura 4.2.

Se tienen las siguientes observaciones:

El rendimiento de las tres formulaciones mejora al aumentar el orden de magnitud del 10 % de datos, como se puede apreciar en la parte (a) de la figura 4.2. Una posible explicación para ello, es que a mayor diferencia de magnitud entre el 10 % de entradas y el resto, más fácil discernir los objetivos prioritarios tanto para el defensor como para el atacante.

En la parte (b) de la figura 4.2 se observa que con un orden de magnitud de 10^2 , (SSGM) y (DOBSS-SSG) son las formulaciones dominantes y que cuando el orden de magnitud sube a 10^3 , la formulación dominante es (DOBSS-SSG).

Por otro lado, en la parte (c) de la figura 4.2 se aprecia que cuando el orden de magnitud es de 10^4 , ya no hay una formulación claramente dominante, en el 60 % de las ejecuciones no

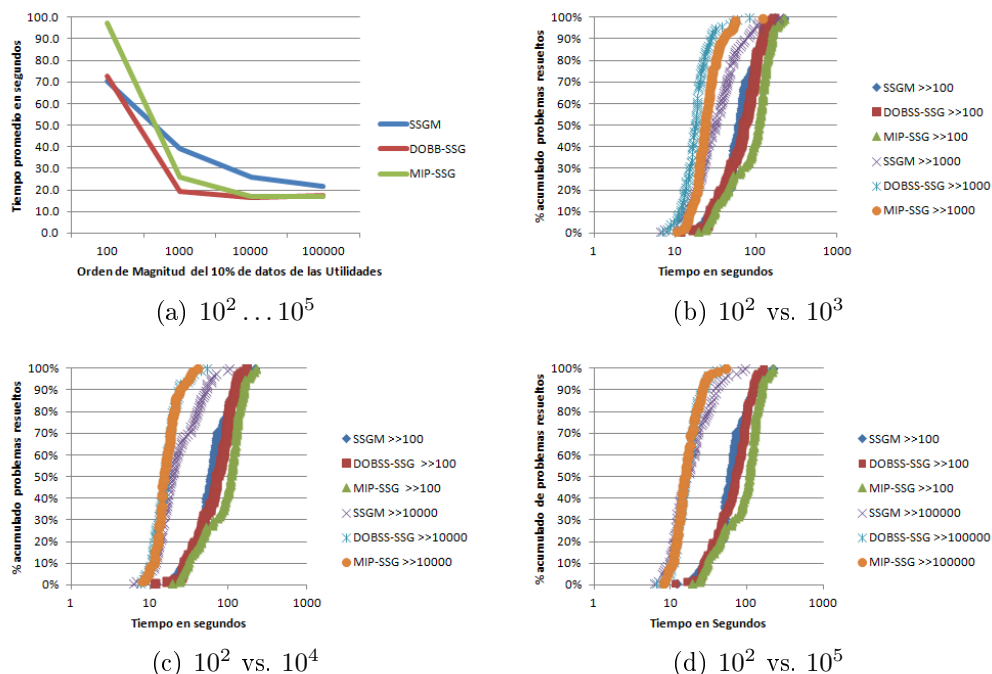


Figura 4.2: Efecto del orden de magnitud del 10 % de de datos.

se distingue un claro ganador entre las tres formulaciones y en el otro 40 % el ganador está entre (DOBSS-SSG) y (MIP-SSG), y (SSGM) no se encuentra extremadamente alejado de ellas.

Finalmente, como se ve en la parte (d) de la figura 4.2, si se llega un orden de magnitud igual a 10^5 , se incrementa ligeramente el efecto observado cuando el orden de magnitud es 10^4 , donde hay un 70 % de ejecuciones donde no se distingue un claro ganador entre las tres formulaciones.

4.4.3. Efecto de la estructura de la matriz de utilidades

En propósito de esta sección es identificar si el efecto del orden de magnitud del 10 % de datos revisado en la sección 4.4.2 esta relacionado con la estructura de la matriz de utilidades y cual es su efecto en las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Para ello, también se toma el set B pero se varía la estructura de la matriz de dos formas: la estructura 1, la original, que elige el 10 % de datos de mayor orden de manera aleatoria; y la estructura 2, que elige aquellos datos correspondientes a una misma locación y, en caso la cantidad de entradas modificadas no alcance el 10 %, se elige otra locación y todos sus

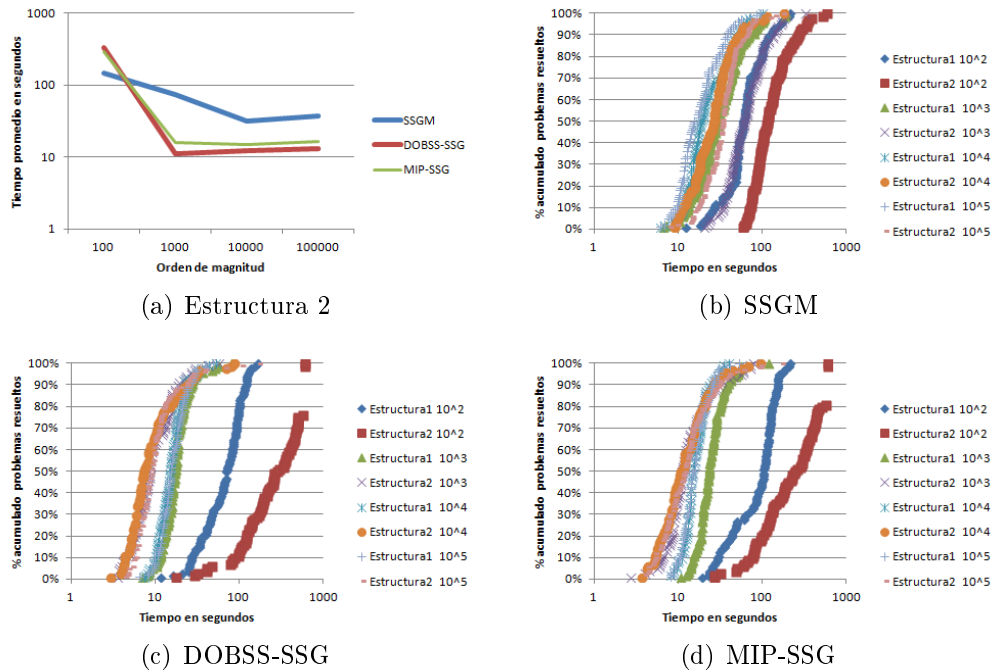


Figura 4.3: Efecto de la estructura de la matriz de utilidades.

entradas correspondientes y así sucesivamente hasta completar el 10 % de datos.

El diseño de este experimento considera un tiempo límite de 600 segundos, 50 ejecuciones por tipo de estructura y por orden de magnitud. El orden de magnitud del 10 % de datos empieza en 10^2 y sube un orden de magnitud a la vez hasta llegar a 10^5 .

Los resultados se muestran en la figura 4.3.

Se tienen las siguientes observaciones:

El efecto del orden de magnitud de los datos se mantiene al variar la estructura de la matriz, ver parte (a) de la figura 4.3.

Como se puede apreciar en la parte (b) de la figura 4.3, la formulación (SSGM) tiene tiempos más lentos con la estructura 2 que con la estructura 1; sin embargo, el efecto del orden de magnitud no depende de la estructura ya que en ambas disminuye el tiempo conforme aumenta el orden de magnitud.

El tipo de estructura 2, aquel que elige el 10 % de las entradas siempre en la misma locación, produce una mayor reducción del tiempo en las formulaciones (DOBSS-SSG) y (MIP-SSG) que en (SSGM) como se puede apreciar al comparar las partes (c) y (d) con la parte (b) de la figura 4.3.

En ambas estructuras 1 y 2, y con un orden de magnitud mayor o igual a 10^3 , las tres formulaciones alcanzan la optimalidad en el 100 % de los problemas resueltos. Mientras que con un orden de magnitud de 10^2 solo (SSGM) alcanza optimalidad en el 100 % de los problemas resueltos, en las otras dos formulaciones (DOBSS-SSG) y (MIP-SSG) el 20 % de las instancias presenta gap con este mismo orden de magnitud.

4.4.4. Efecto del número de tipos de atacantes

El propósito de esta sección es determinar el efecto del número de tipos de atacantes, $|L|$, sobre las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Para ello, se realiza dos experimentos, el primero que establece un tiempo límite para medir el porcentaje de instancias que pueden ser resueltas en un cierto periodo de tiempo, y el segundo que considera la resolución de las instancias sin tiempo límite para apreciar las diferencias absolutas entre las formulaciones.

El diseño del primer experimento usa el caso base, set B en la tabla 4.1 y mantiene todos los parámetros fijos menos $|L|$, el cual varía [$+/- 1$ unidad]. Este experimento considera un tiempo límite de 600 segundos, 50 ejecuciones por variación de $|L|$, fija la estructura de matriz al tipo 1, es decir, elige aleatoriamente el 10 % de datos de mayor magnitud y fija el orden de magnitud igual a 10^4 . Esto último se fundamenta en las observaciones de la sección 4.4.2.

Los resultados se muestran en la figura 4.4.

Se tienen las siguientes observaciones:

La dificultad de resolución aumenta al aumentar $|L|$ para las tres formulaciones.

Al comparar las partes (a) y (b) de la figura 4.4, se puede apreciar que con $|L| = 5$ (SSGM) domina a las otras dos formulaciones en el 85 % de las instancias; sin embargo su rendimiento cae notoriamente cuando $|L|$ sube tan sólo una unidad.

Al aumentar el número de tipos de atacantes las formulaciones (DOBSS-SSG) y (MIP-SSG) tienen un mejor rendimiento que la formulación (SSGM), esto se hace visible cuando $|L| \geq 6$, como se puede apreciar en la parte (c) de la figura 4.4. Sin embargo, no existe un claro ganador entre (DOBSS-SSG) y (MIP-SSG), el experimento descrito a continuación fue diseñado para ayudar a dilucidar si en algún momento una llega a ser mejor que la otra.

El diseño del segundo experimento también usa el caso base, set B en la tabla 4.1 y mantiene todos los parámetros fijos menos $|L|$, el cual varía entre los valores [10... 12]. Este

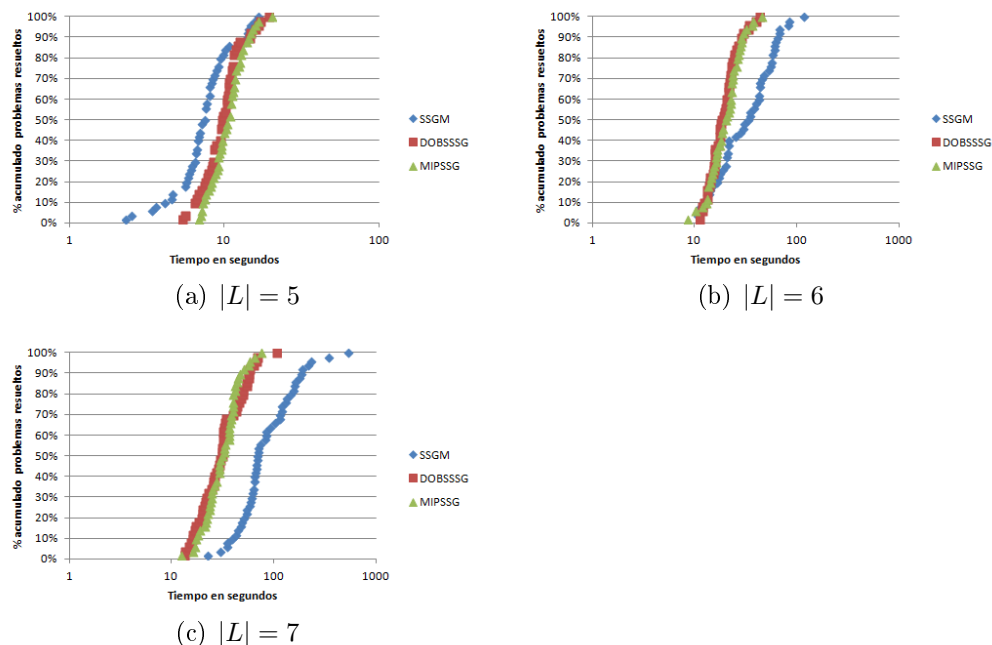


Figura 4.4: Efecto del número de tipos de atacantes $|L|$.

experimento se realiza sin tiempo límite, considera 3 ejecuciones por variación de $|L|$, fija la estructura de matriz al tipo 1, es decir, elige aleatoriamente el 10 % de datos de mayor magnitud y fija este orden de magnitud en 10^2 .

Como se muestra en la figura 4.5, la formulación (MIP-SSG) es la que mejor se comporta cuando $|L|$ toma valores grandes y se mantienen fijos los demás parámetros ($|Q| = 40$ y $m = 0,3|Q|$). En la parte (a) se puede observar que los tiempos de resolución promedio van desde los 1,000 segundos con la formulación (MIP-SSG) (≈ 30 minutos) hasta aproximadamente 30,000 segundos con la formulación (SSGM) (≈ 8 horas) y en la parte (b) se puede apreciar que la cantidad de nodos de branch and bound explota hasta llegar en promedio a más de 10 millones de nodos con (SSGM).

4.4.5. Efecto del número de objetivos

El propósito de esta sección es determinar el efecto del número de objetivos, $|Q|$, sobre las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Para ello, se realiza dos experimentos, el primero que establece un tiempo límite para medir el porcentaje de instancias que pueden ser resueltas en un cierto periodo de tiempo, y el segundo que considera la resolución de las instancias sin tiempo límite para apreciar las

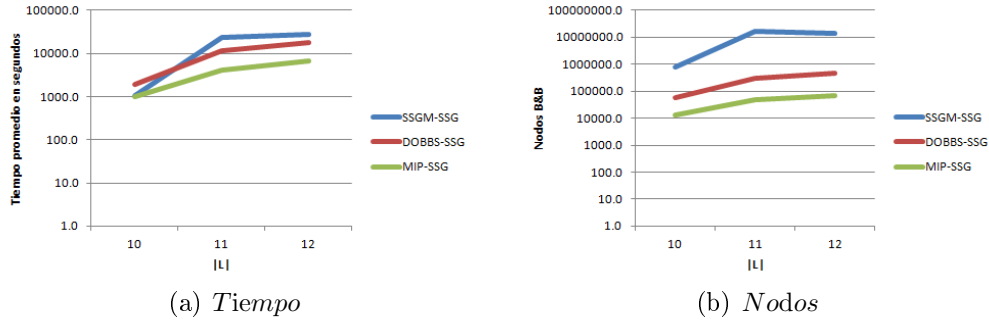


Figura 4.5: Efecto del número de atacantes $|L|$ sin tiempo límite.

diferencias absolutas entre las formulaciones.

El diseño del primer experimento usa el caso base, set B en la tabla 4.1 y mantiene todos los parámetros fijos menos $|Q|$, el cual varía $[+/- 1$ decena]. Este experimento considera un tiempo límite de 600 segundos, 50 ejecuciones por variación de $|Q|$, fija la estructura de matriz al tipo 1, es decir, elige aleatoriamente el 10% de datos de mayor magnitud y fija este orden de magnitud en 10^4 . Esto último se fundamenta en las observaciones de la sección 4.4.2.

Los resultados se muestran en la figura 4.6.

Se tienen las siguientes observaciones:

La dificultad de resolución aumenta al aumentar $|Q|$ para las tres formulaciones.

Al aumentar el número de objetivos las formulaciones (DOBSS-SSG) y (MIP-SSG) mejoran su rendimiento con relación a la formulación (SSGM); sin embargo, no hay un claro ganador entre ambas.

La formulación (SSGM) es la que mejor se desempeña cuando $|Q| = 30$, esto se ve en la parte (a) de la figura 4.6. Sin embargo, cuando $|Q| = 50$ no solo es la que peor rendimiento tiene sino que también tiene un 5% de instancias con gap mientras que las otras dos formulaciones resuelven todas las instancias a optimalidad.

Al aumentar el número de objetivos las formulaciones (DOBSS-SSG) y (MIP-SSG) tienen un mejor rendimiento que la formulación (SSGM), esto empieza a ser visible cuando $|Q| \geq 40$, como se puede apreciar en las partes (b) y (c) de la figura 4.4. Sin embargo, no existe un claro ganador entre (DOBSS-SSG) y (MIP-SSG), el experimento descrito a continuación fue diseñado para ayudar a dilucidar si en algún momento una llega a ser mejor que la otra.

El diseño del segundo experimento usa el caso base, set B en la tabla 4.1 y mantiene todos los parámetros fijos menos $|Q|$, el cual varía entre los valores $[60, 70, 80]$. Este experimento

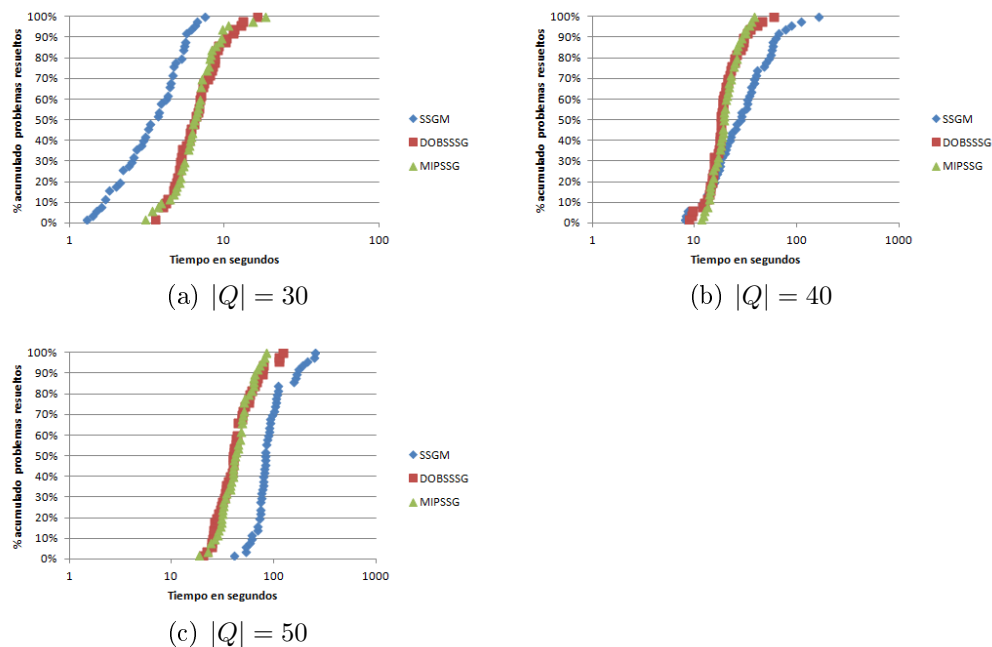


Figura 4.6: Efecto del número de objetivos $|Q|$.

se realiza sin tiempo límite, considera 3 ejecuciones por variación de $|L|$, fija la estructura de matriz al tipo 1, es decir, elige aleatoriamente el 10 % de datos y fija el orden de magnitud de este 10 % de datos en 10^2 .

Como se muestra en la figura 4.7, las formulaciones (DOBSS-SSG) y (MIP-SSG) son las que mejor se comportan cuando $|Q|$ toma valores grandes y se mantienen fijos los demás parámetros en $|L| = 6$ y $m = (0,3|Q|)$. En la parte (a) se puede observar que los tiempos de resolución promedio van desde los 300 segundos con la formulación (DOBSS-SSG) (≈ 5 minutos) hasta casi 23,000 segundos con la formulación (SSGM) (≈ 6 horas) y en la parte (b) se puede apreciar que la cantidad de nodos de branch and bound explota hasta llegar en promedio a más de 26 millones de nodos con (SSGM).

4.4.6. Efecto del número de recursos del defensor

El propósito de esta sección es determinar el efecto del número de recursos de seguridad m sobre las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Para ello, se toma un caso base, set B en la tabla 4.1 y se mantiene todos los parámetros fijos menos m , el cual varía $[+/- (0,1 \cdot |Q|)]$. Es decir, para efecto del análisis se toma valores de m que representan el ratio de protección que proveen los recursos de defensa con respecto

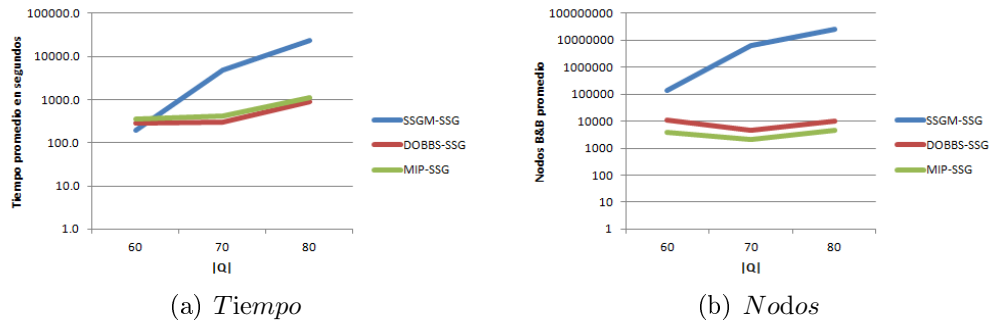


Figura 4.7: Efecto del número de objetivos $|Q|$ sin tiempo límite

al total de objetivos. Por ejemplo, $m = 0,3$ indica que la cantidad de recursos de defensa es capaz de cubrir hasta el 30 % del total de objetivos.

El diseño de este experimento considera un tiempo límite de 600 segundos, 50 ejecuciones por variación de $|m|$. Los resultados se muestran en la figura 4.8.

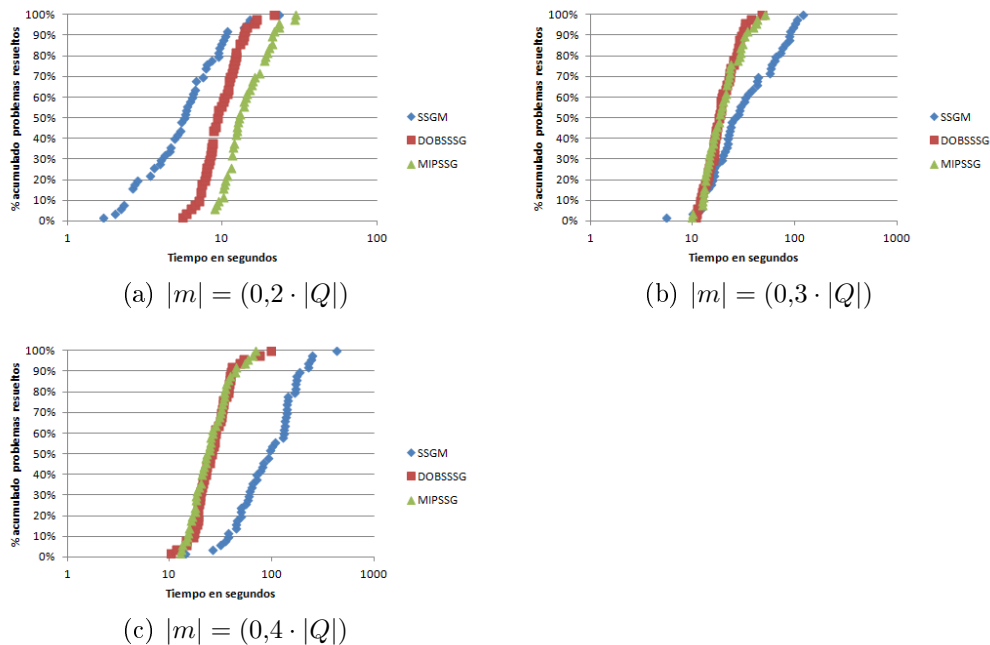


Figura 4.8: Efecto del número de recursos del defensor $|m|$.

Como se puede notar, la complejidad es mayor cuando el ratio es del 40 % ($m = (0,4 \cdot |Q|)$). Por lo tanto, se confirma lo mismo que se detalla en [18], donde se detalla que el ratio de recursos de defensa disponibles sobre el total objetivos es crítico cuando es cercano al 50 %. Por ejemplo, si se observa la formulación (SSGM), en la parte (a) de la figura 4.8, esta formulación resuelve el 100 % de las instancias en menos de 100 segundos, mientras que en

la parte (c) de la misma figura se aprecia que el 50 % de las instancias supera este tiempo.

También se observa que al aumentar la cantidad de recursos de defensa, las formulaciones (DOBSS-SSG) y (MIP-SSG) tienen un mejor rendimiento que la formulación (SSGM); sin embargo, no hay un claro ganador entre ambas.

4.5. Análisis de las formulaciones logarítmicas

Un tema importante a considerar en la implementación de modelos teóricos para resolver casos reales, es analizar el rendimiento de las formulaciones en instancias de gran escala, ya que esto es casi siempre análogo a problemas del mundo real.

Como se explica en la sección 3.4, se espera que las formulaciones logarítmicas ayuden a resolver instancias grandes en menor tiempo. Según [42] la reducción del número de variables binarias ayuda en el tiempo de resolución porque usualmente son estas variables las que suelen generar arboles desbalanceados al momento de realizar el branch and bound, lo que causa a su vez que los solvers del estado del arte, por lo general, sean muy lentos al momento de hallar una solución óptima.

Ante este problema, el propósito de esta sección es determinar el efecto que tiene la representación logarítmica de los objetivos y su escalabilidad, comparando las tres formulaciones logarítmicas (LSSGM), (LDOBSS-SSG) y (LMIP-SSG), modelos (3.24), (3.25) y (3.26), contra las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Para ello, se realiza un experimento que considera la resolución de las instancias sin tiempo límite y así poder apreciar tanto las diferencias absolutas entre las formulaciones como su escalabilidad.

El diseño de este experimento usa el set de instancias D , descrito en la tabla 4.1 y agrega valores de $|L|$ bajos para poder ilustrar el cambio en complejidad de pasar de parámetros bajos a los altos del set D . Este experimento realiza 3 ejecuciones por variación de $|Q|$, fija el tipo de estructura de la matriz al tipo 1, que elige aleatoriamente el 10 % de datos de mayor magnitud, y fija el orden de magnitud igual de este 10 % de datos en 10^2 .

En la tabla 4.5 se muestra las variaciones en el tamaño cuando se usa el modelo (SSGM) versus (LSSGM), la primera formulación sin y con logaritmos respectivamente.

En la tabla 4.6 se muestra las variaciones en el tamaño cuando se usa el modelo (DOBSS-SSG) versus (LDOBSS-SSG), es decir, la segunda formulación sin y con logaritmos respectivamente.

Q	Var		VarI		Const	
	No Log	Log	No Log	Log	No Log	Log
4	16	20	8	4	27	31
8	28	34	16	6	51	57
16	52	60	32	8	99	107
32	100	110	64	10	195	205
64	196	208	128	12	387	399
128	388	402	256	14	771	785
256	772	788	512	16	1539	1555
512	1540	1558	1024	18	3075	3093
1024	3076	3096	2048	20	6147	6167

Tabla 4.5: SSGM: variación del tamaño al incluir la representación logarítmica

Q	Var		VarI		Const	
	No Log	Log	No Log	Log	No Log	Log
4	46	50	8	4	115	119
8	154	160	16	6	419	425
16	562	570	32	8	1603	1611
32	2146	2156	64	10	6275	6285
64	8386	8398	128	12	24835	24847
128	33154	33168	256	14	98819	98833
256	131842	131858	512	16	394243	394259
512	525826	525844	1024	18	1574915	1574933

Tabla 4.6: DOBSS-SSG: variación del tamaño al incluir la representación logarítmica

Q	Var		VarI		Const	
	No Log	Log	No Log	Log	No Log	Log
4	44	48	8	4	131	135
8	152	158	16	6	515	521
16	560	568	32	8	2051	2059
32	2,144	2,154	64	10	8195	8205
64	8,384	8,396	128	12	32771	32783
128	33,152	33,166	256	14	131075	131089
256	131,840	131,856	512	16	524291	524307
512	525,824	525,842	1024	18	2097155	2097173

Tabla 4.7: MIP-SSG: variación del tamaño al incluir la representación logarítmica

En la tabla 4.7 se muestra las variaciones en el tamaño cuando se usa el modelo (MIP-SSG) versus (LMIP-SSG), es decir, la tercera formulación sin y con logaritmos respectivamente.

Como se muestra en las tablas 4.5, 4.6 y 4.7 la formulación logarítmica sí reduce el número de variables binarias; pero no reduce el tamaño. Por ejemplo, considerando la tabla 4.7 con $|Q| = 256$, usando la formulación logarítmica el número de variables binarias se reduce de 512 a sólo 16, pero el tamaño, expresado en número de variables más número de restricciones, pasa de $131840 + 524291 = 656131$ a $131856 + 524307 = 656163$, es decir, quedan prácticamente del mismo tamaño.

La figura 4.9 visualiza el efecto de las formulaciones logarítmicas (LSSGM), (LDOBSS-SSG) y (LMIP-SSG) sobre las variables enteras. El comportamiento es el mismo para las tres formulaciones.

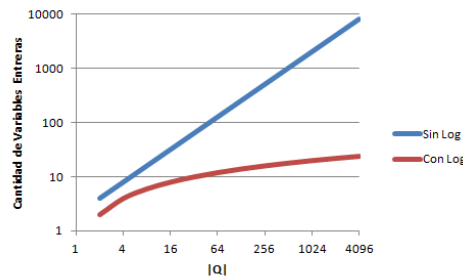


Figura 4.9: Efecto de la formulación logarítmica en el número de variables enteras.

Como se menciona en la sección 3.4, se espera que las formulaciones logarítmicas (3.24) (3.25) y (3.26) ayuden a resolver instancias de gran tamaño en menor tiempo, para verificar esto se compara los tiempos de ejecución de estas formulaciones versus las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

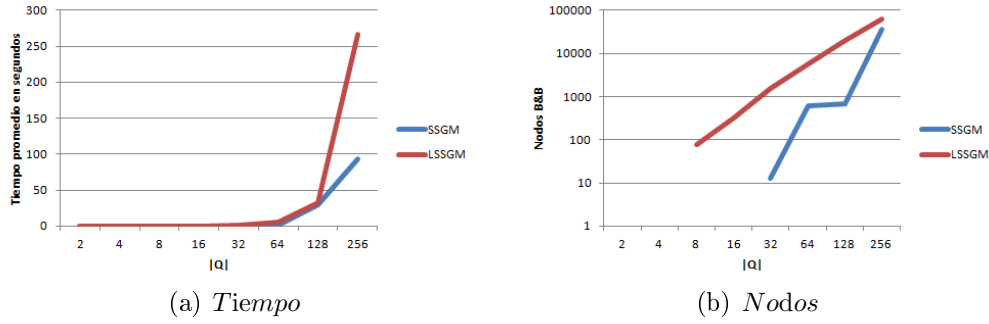


Figura 4.10: Perfil de las formulaciones SSGM versus LSSGM.

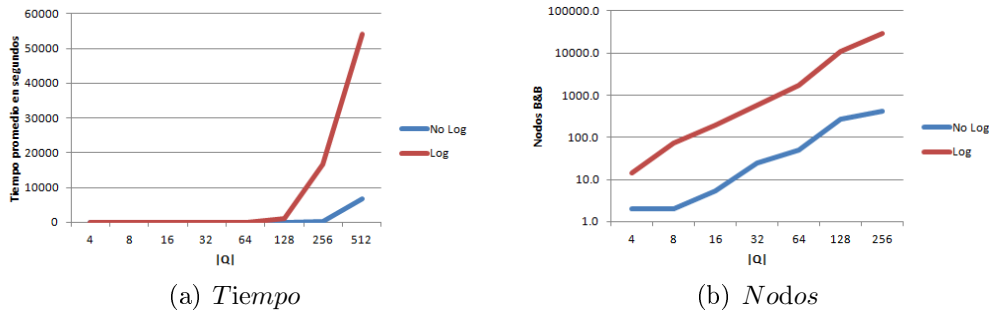


Figura 4.11: Perfil de las formulaciones DOBSS-SSG versus LDOBSS-SSG.

En las figuras 4.10, 4.11 y 4.12 se muestra los perfil de rendimiento comparando las formulaciones mencionadas.

Se tiene las siguientes observaciones:

Por lo general el comportamiento de las formulaciones logarítmicas no ayuda a disminuir el tiempo de resolución. En la parte (a) de las figuras 4.10, 4.11, y 4.12 cuando $|Q| = 256$ se aprecia que (LSSGM) es tres veces más lenta que (SSGM), (LDOBSS-SSG) es diez veces más que (DOBSS-SSG) y (LMIP-SSG) es veinticinco veces más lenta que (MIP-SSG).

En el caso de las formulaciones (LDOBSS-SSG) y (LMIP-SSG) las ejecuciones fueron interrumpidas por falta de memoria al llegar a 512 objetivos y $|L| = 2$.

Las formulaciones (DOBSS-SSG) y (MIP-SSG) fueron interrumpidas por falta de memoria al llegar a 1024 objetivos y $|L| = 2$.

Mientras que las formulaciones (SSGM) y (LSSGM) que lograron resolver instancias con 1024 objetivos y $|L| = 2$ en aproximadamente 4 y 8 horas, respectivamente.

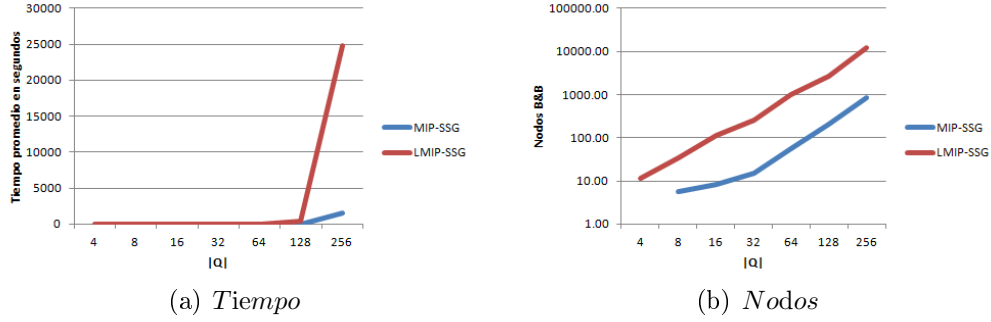


Figura 4.12: Perfil de las formulaciones MIP-SSG versus LMIP-SSG.

4.6. Análisis de la restricción del espacio de soluciones del defensor

En todas las secciones previas a esta, se considera que el espacio de soluciones del defensor es irrestricto, es decir, se considera que todas las posibles combinaciones de asignaciones de recursos a objetivos son factibles. Sin embargo, en las aplicaciones del mundo real esto no siempre es posible.

Por ello, en esta sección se busca limitar las estrategias del defensor en base a dos criterios: el primero considera que solo se puede asignar recursos a locaciones consecutivas, y el segundo considera que un subconjunto de las locaciones podría tener locaciones prohibidas, en el sentido que no pueden ir juntas en una misma estrategia.

4.6.1. Efecto de restringir a locaciones consecutivas

Este escenario fue pensado considerando limitaciones operacionales como el del ámbito descrito en [12] donde se debe tener en cuenta las condiciones de elevación y la distancia a un centro base para asignar los agentes a las zonas de prevención de caza furtiva. Con esta motivación se usa una definición alterna de adyacencia y se define a dos locaciones como consecutivas siempre y cuando estas estén a distancias similares de un punto de referencia o que tengan una distancia similar entre ambas como se explica en la sección 3.5.1.

Ante este escenario, el propósito de esta sección es analizar el efecto de incorporar este tipo de restricciones a las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.28), (3.29) y (3.30) respectivamente.

Para ello, el diseño del experimento considera el set A de forma parcial, utilizando los siguientes parámetros: $|L| = [2, 4, 6]$, $|Q| = [10, 30, 50]$, y $m = [(0,3|Q|), (0,5|Q|), (0,7|Q|)]$.

Realiza 10 ejecuciones por set de parámetros $|L|$, $|Q|$ y m , se fija una matriz de utilidades con una estructura que elige aleatoriamente el 10% de entradas con un orden mayor de magnitud y fija este orden de magnitud en 10^2 .

La figura 4.13 muestra el tiempo de resolución de las tres formulaciones restringidas a locaciones consecutivas, muestra la comparación entre estas tres y también muestra la comparación de cada formulación versus el caso irrestricto.

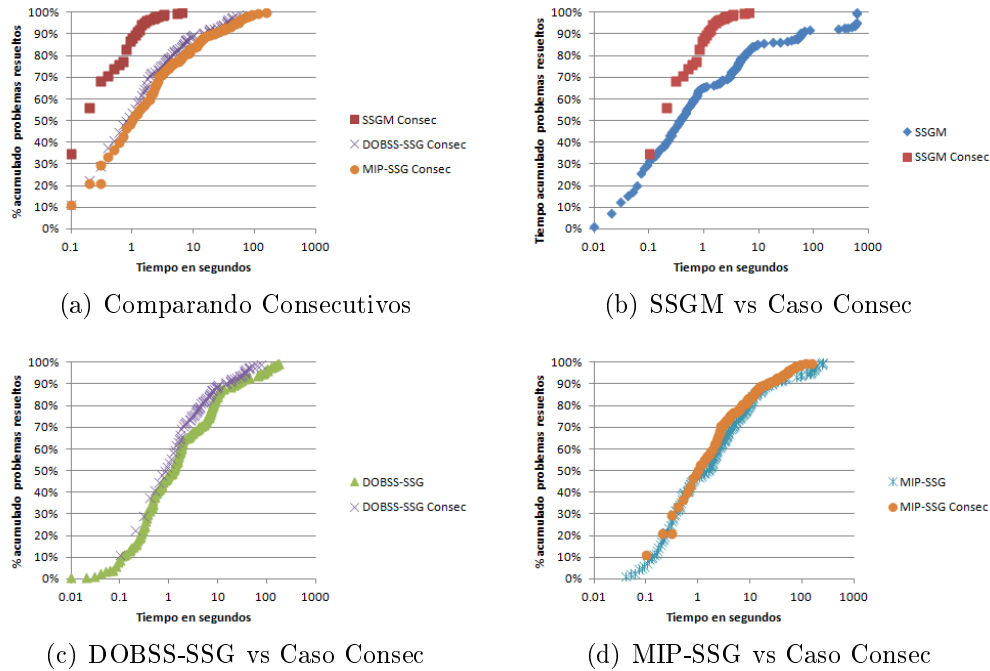


Figura 4.13: Tiempos de ejecución en el caso de restricción a objetivos consecutivos.

Se tienen las siguientes observaciones:

Como se observa en la parte (a) de la figura 4.13, la formulación (SSGM) es la dominante.

En el caso irrestricto con $|L| \geq 6$ el rendimiento de (SSGM) decaía y tanto (DOBSS-SSG) como (MIP-SSG) la superaban, ver sección 4.4.1; sin embargo, usando locaciones consecutivas este comportamiento cambia y (SSGM) pasa a dominar incluso en aquellas instancias con $|L| = 6$.

Como se observa en la parte (b) de la figura 4.13, el impacto de la restricción a consecutivos es muy favorable para (SSGM); mientras que en las partes (c) y (d) se observa que las formulaciones (DOBSS-SSG) y (MIP-SSG) conservan el mismo rendimiento en ambos casos.

A modo de detalle, la tabla 4.8 presenta la comparación de los tiempos promedio de ejecución del caso con restricción a locaciones consecutivas y el caso sin restricciones cuando

$|L| = 2$ y $|L| = 6$.

$ L $	$ Q $	SSGM		DOBSSG		MIPSSG	
		Consec	Sin Restricciones	Consec	Sin Restricciones	Consec	Sin Restricciones
2	10	0.10	0.02	0.12	0.08	0.12	0.09
2	30	0.15	0.07	0.54	0.37	0.61	0.50
2	50	0.29	0.50	1.40	1.25	1.83	1.67
6	10	0.15	0.25	0.30	0.53	0.34	0.41
6	30	0.74	3.77	4.27	6.74	7.64	8.96
6	50	1.71	404.38	24.91	75.22	44.27	106.68

Tabla 4.8: Locaciones consecutivas vs caso irrestricto: tiempos de resolución.

Por ejemplo, si comparamos el caso cuando $|L| = 2$ y $|Q| = 30$ con el caso cuando $|L| = 2$ y $|Q| = 30$, se puede observar que con $|L| = 2$ todas las formulaciones del caso irrestricto son más rápidas que el caso de consecutivos; mientras que con $|L| = 6$ es lo contrario y el caso restringido domina al caso irrestricto. Una posible explicación para esto es que el espacio de soluciones del caso restringido es menor que el del caso irrestricto ($|Q| - m + 1$ versus $|Q|$) lo que podría estar dando ventajas de resolución en instancias de mayor tamaño.

4.6.2. Efecto de definir locaciones con prohibiciones

El propósito de esta sección es determinar el efecto que tiene definir restricciones a locaciones en el rendimiento de las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG), modelos (3.13), (3.18) y (3.21) respectivamente.

Tal como se explica en la sección 3.5.2 se considera que ciertos objetivos no pueden ir en la misma estrategia junto a los objetivos que le estan prohibidos. Estas prohibiciones pueden ser originadas por limitaciones operacionales, tales como dificultad de acceso a ciertas regiones, entre otros.

Para ello, este experimento se utiliza el set de instancias A limitado a $|L| = 2$ y $|Q| = 10$. Se elige las instancias más chicas porque en la generación del espacio de estrategias factibles primero se tiene que generar todas las posibles combinaciones para luego eliminar las prohibidas, este proceso previo limita la escalabilidad del experimento.

El diseño de este experimento considera un tiempo límite de 600 segundos, 10 ejecuciones por set de parámetros $|L|$ y $|Q|$, usa una matriz generada aleatoriamente con una estructura que fija de forma aleatoria el 10 % de los datos de mayor magnitud y fija este orden de magnitud en 10^2 .

Para generar las locaciones prohibidas primero se elige de manera aleatoria un subconjunto $I \subset Q$, cada objetivo que pertenezca a I presentará incompatibilidad con algún otro

objetivo. Luego, se define el conjunto $P(j)$, $\forall j \in I$, que contiene los objetivos prohibidos para cada elemento de I , la selección de estas locaciones prohibidas también se realiza de manera aleatoria. En este experimento, el tamaño de I se fija en $\frac{|Q|}{5}$ y $\frac{|Q|}{3}$ y el tamaño del conjunto $P(j)$ varía de 1 a 3. Con estos dos conjuntos definidos, se procede a eliminar aquellas estrategias que contengan a $j \in I$ y a $p \in P(j)$ al mismo tiempo. Esto conlleva a una reducción del espacio de soluciones.

En la figura 4.14 se muestra como se desempeñan las formulaciones conforme aumenta la cantidad total de objetivos prohibidos.

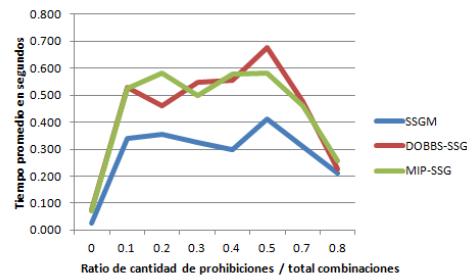


Figura 4.14: Efecto de de definir objetivos con prohibiciones.

Se tienen las siguientes observaciones:

El tiempo de resolución es mucho más bajo si no existieran restricciones al espacio de soluciones del defensor.

Se mantiene el comportamiento de las formulaciones con respecto a lo analizado en la sección 4.4.1. La formulación (SSGM) es la que mejor se comporta al tener instancias chicas tanto en casos irrestrictos como en el caso de objetivos con prohibiciones.

El punto crítico de rendimiento de las tres formulaciones se alcanza cuando la cantidad total de estrategias prohibidas alcanza un ratio de 0,5 con relación al número total de combinaciones.

Al prohibir demasiadas combinaciones, ratio 0.8 en la figura 4.14, las tres formulaciones tienen el mismo desempeño y resuelven las instancias rápidamente, esto debido a que el espacio de soluciones es pequeño. Notar que el ratio 0.8 significa que el espacio de soluciones con prohibiciones es 80 % menor que el espacio de soluciones sin prohibiciones,

Conclusión

En esta tesis se han revisado tres formulaciones de programación entera mixta propuestas para el problema de asignación de recursos de seguridad usando los Juegos de Seguridad de Stackelberg (SSG) y se ha discutido dos extensiones: la inclusión de una representación logarítmica de las variables enteras y la restricción del espacio de soluciones factibles para el defensor.

A través de pruebas computacionales, se ha mostrado cómo estas formulaciones se desempeñan en problemas de tamaño mediano a grande. Es importante tener en cuenta que debido a que el tamaño de las instancias es limitado a parámetros específicos, los resultados aquí presentados deben ser tomados como ilustrativos en vez de significativos estadísticamente en sus conclusiones.

Todas estas formulaciones, que han sido desarrolladas durante la última década, constituyen un buen paso para resolver los SSG a optimalidad; sin embargo, desde el análisis aquí presentado se ve que todavía hay lugar a mejora.

En la tabla 4.3 se ve que las tres formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG) dan todas cotas de relajación lineal poco ajustadas. Sin embargo, en problemas de menor tamaño es posible resolver estas formulaciones via branch and bound de manera rápida, ver cuadro 4.4.

Como se muestra en la sección 4.2, tal cual se esperaba, las formulaciones (DOBSS-SSG) y (MIP-SSG) resultan en problemas mucho más grandes que (SSGM) en términos de número de variables y número de restricciones, y son también mucho más densas en términos de no-ceros. Esto se debe a que (SSGM) usa la técnica de la gran M y además no usa restricciones lógicas como las usadas en las otras dos formulaciones. Por lo general conforme más grande es el problema de programación entero mixto, más difícil es de resolver, principalmente porque requiere una cantidad grande de memoria para almacenar información.

Sin embargo, como se muestra en la sección 4.3, la formulación (MIP-SSG), la de tamaño más grande, es como se esperaba la formulación más ajustada de las tres y, como se muestra en la sección 4.4, es justo (MIP-SSG) la formulación con el mejor rendimiento cuando las instancias son de dificultad alta, sobretodo instancias con $|L| \geq 7$, ver figura 4.4. Esto se

cumple siempre y cuando $|Q|$ no sea excesivamente grande ya que por un tema de tamaño podría faltar memoria para resolver el problema.

Un tema importante a considerar en la implementación de modelos teóricos para resolver casos reales, es analizar el rendimiento de las formulaciones en instancias de gran escala, ya que esto es casi siempre análogo a problemas del mundo real.

Entonces, escalando la cantidad de tipos de atacantes, en la figura (4.5) se puede apreciar que en problemas con 12 tipos de atacantes, 40 objetivos y con una cantidad de recursos de defensa capaz de cubrir hasta el 30 % del total de objetivos, (MIP-SSG) es capaz de resolver a optimalidad en 1.86 horas, llevando la delantera a (DOBSS-SSG) y a (SSGM) que resuelven en 5.05 horas y 7.67 horas respectivamente.

Por otro lado, escalando la cantidad de objetivos a proteger, en la figura (4.7) se puede apreciar que en problemas con 6 tipos de atacantes, 80 objetivos a proteger y con una cantidad de recursos de defensa capaz de cubrir hasta el 30 % del total de objetivos, (MIP-SSG) es capaz de resolver a optimalidad en 18.95 minutos y (DOBSS-SSG) resuelve en 14.99 minutos, ambas llevan la delantera a (SSGM) que recién llega a optimalidad en 6.39 horas.

Al comparar estos resultados con los rendimientos de las formulaciones de la literatura se puede notar que en el caso de un gran número de tipos de atacantes, el rendimiento de (MIP-SSG) supera a los resultados obtenidos previamente. Por ejemplo, (DOBSS) propuesto por [30] y usado para el problema del aeropuerto de Los Ángeles, no es capaz de resolver problemas de los tamaños indicados en los dos párrafos anteriores, ya que se queda sin memoria. (ERASER) propuesto por [25] puede escalar hasta 3000 objetivos con 1 sólo tipo de atacante en 13.30 minutos pero los autores en dicho trabajo no presentan resultados para más de 1 tipo de atacante. Sin embargo, el autor de [19] sí muestra resultados para (ERASER) con 2 tipos de atacantes, fijando la cantidad de objetivos en 50 y considerando una capacidad de cobertura del 30 % y obtiene un tiempo de resolución de 0.80 segundos; tomando estos mismos parámetros (SSGM), (DOBSS-SSG) y (MIP-SSG) resuelven a optimalidad en 0.27, 1.63 y 2.30 segundos respectivamente. En este mismo trabajo este autor propone dos formulaciones para el caso de seguridad, (ASPEN) bayesiano y (HBSA), y escala la cantidad de tipos de atacantes hasta un máximo de 6 tipos de atacantes, fija la cantidad de objetivos en 50 y considera que la cantidad de recursos de defensa es capaz de cubrir sólo el 10 % del total de objetivos, con estos parámetros el autor indica que ninguna de estas dos formulaciones logra resolver a optimalidad con 6 tipos de atacantes en menos de 24 horas y que con 5 tipos de atacantes alcanza optimalidad en un tiempo superior a 16 horas, lo que en escenarios de la vida real resulta ser prohibitivo ya que las agencias de seguridad requieren asignar recursos en tiempo real. Considerando estos resultados, las formulaciones estudiadas en este trabajo presentan una mayor ventaja al escalar a problemas grandes, sobretodo al escalar la cantidad de tipos de atacantes.

Otro tema importante es evaluar la sensibilidad de los parámetros de las formulaciones

(SSGM), (DOBSS-SSG) y (MIP-SSG), para ello se tomó un caso base, ver parámetros del set B de la tabla 4.1, y se tiene las siguientes observaciones:

- El orden de magnitud del 10 % de entradas de la matriz de utilidades sí influye en su comportamiento ya que al aumentar de magnitud las formulaciones (DOBSS-SSG) y (MIP-SSG) se ven favorecidas, ver sección 4.4.2.
- Los cambios en la estructura de las matrices, específicamente pasar de tener el 10 % de datos en posiciones aleatorias a posiciones en una misma locación, ayuda a que las tres formulaciones sean resueltas en menor tiempo; pero no alteran el comportamiento observado previamente en las tres formulaciones, ver sección 4.4.3.
- En el caso de los parámetros cantidad de tipos de atacantes $|L|$, cantidad de objetivos $|Q|$ y cantidad de recursos de defensa m , se observa que la dificultad de resolución aumenta al aumentar $|L|$ y $|Q|$, y cuando m es cercano a $0,5|Q|$.

Así mismo, en la sección 4.4 se realizaron una serie de experimentos con instancias grandes con el fin de caracterizar aquellos problemas bajo los cuales una formulación podía ser más ventajosa que otra y se obtuvieron las siguientes observaciones:

- La formulación (MIP-SSG) por lo general es la de mejor rendimiento cuando las instancias tienen muchos atacantes $|L| \geq 11$ y los objetivos son entre $40 \leq |Q| \leq 128$; mientras que (SSGM) por lo general lleva la delantera en instancias con $|L| = 2$ y $|Q| \geq 4$ pudiendo llegar a resolver con 1024 objetivos.
- La formulación (DOBSS-SSG) suele tener un rendimiento similar a (MIP-SSG) cuando $6 \leq |L| \leq 10$ y $40 \leq |Q| \leq 128$, ambas llevan la delantera pero no se distingue un claro ganador entre ambas con estos parámetros.
- La formulación (SSGM) es la que más se perjudica cuando $|L| \geq 6$.

Una de las extensiones a los modelos fue incluir una representación logarítmica de las variables enteras con la expectativa de reducir el tiempo de ejecución y así poder escalar aún más las formulaciones, para ello se comparó las formulaciones (LSSGM), (LDOBSS-SSG) y (LMIP-SSG) contra las formulaciones no-logarítmicas (SSGM), (DOBSS-SSG) y (MIP-SSG). Este análisis se presenta en la sección 4.5, donde se concluye que al incorporar la representación logarítmica de los objetivos no se logra tener un efecto positivo en la escalabilidad de las formulaciones del caso de seguridad. En esta misma sección se detalla hasta donde escala cada una de las formulaciones logarítmicas y se puede apreciar que, en los experimentos realizados, estas tienen un rendimiento muy por debajo de las formulaciones (SSGM), (DOBSS-SSG) y (MIP-SSG).

Según [42] no siempre se logra una formulación más fuerte / rápida al usar la proposición 3.1 y se señala que por lo general se logra una formulación ideal con la representación logarítmica de las variables q si además se elimina esta variable de la formulación. En las tres formulaciones de interés, modelos (3.24), (3.25) y (3.26) se tiene la reducción del número de variables de $|L| \cdot |Q|$ a $\lceil \log_2(|Q|) \rceil$ pero no se obtiene una disminución del tiempo

de resolución, por lo que es necesario replantear las formulaciones logarítmicas para lograr un impacto positivo en la escalabilidad. Una posible extensión al presente trabajo es reformular estos modelos para que al incluir la representación logarítmica de las variables enteras se logre una disminución del tiempo de resolución en el caso de seguridad, esto quedaría como trabajo futuro.

Otra extensión evaluada fue restringir el espacio de soluciones del defensor. En todas las secciones previas a la sección 4.6, se considera que el espacio de soluciones del defensor es irrestricto, es decir, se considera que todas las posibles combinaciones de asignaciones de recursos a objetivos son factibles. Sin embargo, en las aplicaciones del mundo real esto no siempre es posible, el caso de (FAMS) es un claro ejemplo de que no siempre todas las combinaciones son factibles, ver [40].

Por ello, en este trabajo se estudia el efecto de incorporar posibles restricciones al espacio de soluciones del defensor en las tres formulaciones de interés y se toma dos criterios: el primero considera que solo se puede asignar recursos a locaciones consecutivas, y el segundo considera que un subconjunto de las locaciones tiene locaciones prohibidas, en el sentido que no pueden ir juntas en una misma estrategia. Ambos criterios fueron pensados considerando limitaciones operacionales como dificultad de acceso.

En el caso de locaciones consecutivas, la tabla 4.8 y la figura 4.13 muestran que el comportamiento de las formulaciones cambia con respecto al caso irrestricto y se tiene como claro ganador a la formulación (SSGM). En este nuevo escenario ya no se nota una ventaja de (DOBSS-SSG) ni de (MIP-SSG) al aumentar $|L|$ como se observa en la sección 4.4.1.

Por otro lado, en el caso de locaciones con prohibiciones, ver figura 4.14, se tiene que la incorporación de prohibidos también incrementa el tiempo de resolución con respecto al caso irrestricto pero en este caso el comportamiento de las formulaciones entre sí permanece invariable con respecto a lo observado en la sección 4.4.1 en instancias con $|L| = 2$.

Como se ha notado a lo largo de todo el trabajo de tesis, identificar aquella formulación que se desempeña mejor depende fuertemente de la estructura específica del problema y de los datos. Gracias a que existe una alta correlación entre ciertas propiedades de una formulación y un buen rendimiento computacional, en este trabajo se ha logrado identificar aquellas propiedades que impactan en el desempeño de las formulaciones de interés de este trabajo. Todas las observaciones antes mencionadas sirven para mejorar el entendimiento de las formulaciones basadas en teoría de juegos, lo que resulta crítico para continuar desarrollando aplicaciones en dominios del mundo real, como aquellas que enfrenta Carabineros de Chile en sus labores de prevención.

Bibliografía

- [1] Warren P Adams and Stephen M Henry. Base-2 expansions for linearizing products of functions of discrete variables. *Operations research*, 60(6):1477–1490, 2012.
- [2] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- [3] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
- [4] Vicki M Bier. Choosing what to protect. *Risk Analysis*, 27(3):607–620, 2007.
- [5] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel Procaccia, and Arunesh Sinha. Audit games with multiple defender resources. *arXiv preprint arXiv:1409.4503*, 2014.
- [6] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel D Procaccia, and Arunesh Sinha. Audit games. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 41–47. AAAI Press, 2013.
- [7] Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973.
- [8] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, pages 861–898, 2004.
- [9] Vincent Conitzer and Dmytro Korzhyk. Commitment to correlated strategies. In *AAAI*, 2011.
- [10] 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.
- [11] Carabineros de Chile. Portal Institucional de Carabineros de Chile, ENUSC 2015,

- <http://www.carabineros.cl/GestionOperativa/pdf/INFORME.pdf>, último acceso 2015-12-24 enusc, 2015.
- [12] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, and Milind Tambe. Deploying paws to combat poaching: Game-theoretic patrolling in areas with complex terrain. 2015.
 - [13] M Hamisi. *Identification and mapping risk areas for zebra poaching: A case of Tarangire National Park, Tanzania*. PhD thesis, Thesis, ITC, 2008.
 - [14] John C Harsanyi and Reinhard Selten. A generalized nash solution for two-person bargaining games with incomplete information. *Management Science*, 18(5-part-2):80–106, 1972.
 - [15] William Haskell, Debarun Kar, Fei Fang, Milind Tambe, Sam Cheung, and Lt Elizabeth Denicola. Robust protection of fisheries with compass. In *Innovative Applications of Artificial Intelligence (IAAI)*, 2014.
 - [16] Stephen Henry. Tight polyhedral representations of discrete sets using projections, simplices, and base-2 expansions. 2011.
 - [17] Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.
 - [18] Manish Jain, Kevin Leyton-Brown, and Milind Tambe. The deployment-to-saturation ratio in security games. *Target*, 1(5):5, 2012.
 - [19] Manish Jain, Milind Tambe, Vincent Conitzer, Fernando Ordóñez, Bhaskar Krishnamachari, and Mathew McCubbins. Scaling up security games: Algorithms and applications. 2012.
 - [20] 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.
 - [21] Sushil Jajodia, Anup K Ghosh, VS Subrahmanian, Vipin Swarup, Cliff Wang, and X Sean Wang. *Moving Target Defense II*. Springer, 2013.
 - [22] Jara Juan. Modelo de asignación de recursos policiales en la vía pública. Tesis, Universidad de Chile, 2011.
 - [23] Ralph L Keeney. Modeling values for anti-terrorism analysis. *Risk Analysis*, 27(3):585–596, 2007.

- [24] JEJ Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, pages 703–712, 1960.
- [25] 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.
- [26] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, 2010.
- [27] Nupul Kukreja, William GJ Halfond, and Milind Tambe. Randomizing regression tests using game theory. In *Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on*, pages 616–621. IEEE, 2013.
- [28] Han-Lin Li and Hao-Chun Lu. Global optimization for generalized geometric programs with mixed free-sign variables. *Operations research*, 57(3):701–713, 2009.
- [29] George L Nemhauser and Laurence A Wolsey. Integer and combinatorial optimization. *John Wiley & Sons*, 1988.
- [30] Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, 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.
- [31] Praveen Paruchuri, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Security in multiagent systems by policy randomization. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 273–280. ACM, 2006.
- [32] PazCiudadana. Portal Institucional de Paz Ciudadana, Evaluación de la Labor Policial Preventiva, <http://www.pazciudadana.cl/wp-content/uploads/2011/05/evaluacion-labor-policial.pdf>, último acceso 2015-12-24 evaluación policial, 2011.
- [33] 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.

- [34] James Pita, Manish Jain, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Using game theory for los angeles airport security. *AI Magazine*, 30(1):43, 2009.
- [35] James Pita, Milind Tambe, Christopher Kiekintveld, Shane Cullen, and Erin Steigerwald. Guards: innovative application of game theory for national airport security. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2710, 2011.
- [36] Heather Rosoff and Richard John. Decision analysis by proxy for the rational terrorist. In *Quantitative risk analysis for security applications workshop (QRASA) held in conjunction with the International Joint Conference on AI*, pages 25–32. Citeseer, 2009.
- [37] Sui Ruan, Candra Meirina, Feili Yu, Krishna R Pattipati, and Robert L Popp. Patrolling in a stochastic environment. Technical report, DTIC Document, 2005.
- [38] 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.
- [39] David C Sommer. Computational experience with the ophelie mixed integer code. In *talk presented at the International TIMS Conference, Houston*, 1972.
- [40] Jason Tsai, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Shyamsunder Rathi. Iris-a tool for strategic security allocation in transportation networks. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems - Industry Track*, 2009.
- [41] Juan Pablo Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57, 2015.
- [42] Juan Pablo Vielma and George L Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1-2):49–72, 2011.
- [43] H Paul Williams. Experiments in the formulation of integer programming problems. In *Approaches to Integer Programming*, pages 180–197. Springer, 1974.
- [44] Lucy Winchester, Gonzalo Zambrano, and Juan Le-Bert. Informe final de evaluación programa plan cuadrante de seguridad preventiva ministerio de defensa subsecretaría de carabineros. 2007.
- [45] 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.
- [46] Rong Yang, Albert Xin Jiang, Milind Tambe, and Fernando Ordonez. Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 404–410. AAAI Press, 2013.
- [47] 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.
- [48] Zhengyu Yin, Albert Xin Jiang, Milind Tambe, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, and John P Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4):59, 2012.